# 1 Program installation notes

## 1.1 Download

See the Github website

## 1.2 Installation

See the Github website

## 1.3 Program start

**Program does not start due to insufficient rights**

Do such problems occur?

**Program does not start or exits with a runtime error message**

Do such problems occur?

**Note about linking project files (.txp) to UncertRadio.exe**

Will this be used?

**Configuration file UR2\_cfg.dat:**

|  |  |
| --- | --- |
| [UncertRadio configuration]  [Path]  Help\_path=UR2\_CHM/  log\_path=log/  results\_path=results/  example\_path=pros/  [Local]  Decimal\_point=.  List\_separator=,  Language=EN  Monitor#=1  ContrastMode=F | modifiable folder names (relative to UncertRadio.exe):  folder containing the CHM Help file (***local drive!!***)  folder for (temporary) output  folder for output of result files  folder holding the provided example project files  If the contrast mode of the program windows is needed temporarily, it can also be activated in the Options dialog. |

**Country specific parameters**

The **user** of UncertRadio can be **guided** by the program through its various dialogs **in the necessary language**, which can be chosen between German, English and French. This is controlled via the entry ’language=’ in the above-mentioned configuration file UR2\_cfg.dat, which is read at every program start. The parameter values are:

DE for German comma as decimal point; colon as list separator

EN for English point as decimal point; comma as list separator

FR for French comma as decimal point; colon as list separator

The language can be switched within the running program with using the dialog “Menu – Options – Pre-settings“. From the language defined there, the characters for decimal point and list separator are determined. Additionally, the list separator character can there be selected there explicitly.

For a graphical presentation the decimal point character is determined by the language shortcut DE, EN or FR; a manual change of the country/language within Windows is not necessary.

The Windows help can be started also separately from the program itself by double clicking on its filename UR2\_help\_en.chm.

**Configuration file settings.ini:**

[Settings]

gtk-theme-name = win64

gtk-font-name = Sans 11

In this file related to the GTK3-GUI only the entry

gtk-font-name = Sans 11

is allowed to be modified, which defines the font-type and its size.

**Using several monitors**

Working with more than one monitor has become more attractive. In this case, the screens range of coordinates covers all corresponding monitor coordinate ranges. The screen is considered as an envelope rectangular region covering all individual monitor coordinate rectangles. The attribution to a single monitor rectangle is given by the screen-related coordinates of the monitor. A coordinate x is attributed to the width and a coordinate y the height. x=0 defines the left limit and y=0 the upper limit of the screen rectangle; the two coordinate axes are directed from left to right (x) and from top to bottom (y). Within the program, a monitor rectangle is constructed from its upper-left corner coordinates and the sizes of its width and height.

The following Figure shows two possible screen/monitor configurations.

|  |  |
| --- | --- |
| Ein Bild, das Uhr enthält.  Automatisch generierte Beschreibung | screen:  2560 x 1656  (Monitor 2 nicht aktiv) |
|  |  |
| Ein Bild, das Screenshot enthält.  Automatisch generierte Beschreibung | screen:  5760 x 1200 |
|  |  |

In UncertRadio, the screen size, the number of monitors and their rectangle coordinates are determined by GDK functions during the initial starting phase. The values obtained can be found in the file fort66.txt, given in the following form (shown here for only several monitors).

------------------------------------------------------------------------------

coordinates: width x height

\*\*\* Screen: 3200 x 1776

PixelxZoom=120 PixelyZoom=120

\*\*\* Monitors:

\*\*\* 1 Ranges (geom): (1920 - 2560) x (0 - 480) scaling fact= 1.25

\*\*\* 1 Ranges (geom): (0 - 1920) x (480 - 1680) scaling fact= 1.25

\*\*\* 1 Ranges (geom): (1920 - 2560) x (0 - 480) scaling fact= 1.25

\*\*\* # of monitors: 3

\*\*\* Monitor number selected as given in UR2\_cfg.dat: 2

\*\*\* Primary monitor # = 1

\*\*\* Selected monitor: 1; Screen min-max horiz.: 1 - 1919 min-max vertical: 482 - 1642

(und etwas weiter unten)

\*\*\* Main window: first Show: upper-left pos: mposx,mposy= 96 532

\*\*\* Main window: Monitor# at mposx+10,mposy+10= 1

\*\*\* Main window: width= 983 height= 713

------------------------------------------------------------------------------

There are still two problems.

The width and height values taken from the table above, can include a scaling factor being set in Windows 10, e.g., 1.25 for a Windows 10 scaling of 125 %. UncertRadio‘s determination of this factor from Windows 10 may not be safe by now. The scaling in Windows is based on a value of 96 DPI (dots per inch), which is taken there to define a 100 % scaling. If a user has selected e.g. a scaling of 125 %, this is stored as “LogPixels“ within the Windows registry as a value being smaller by a factor 0.96, i.e. 120 %.

Furthermore, the monitor numbers attributed by UncertRadio can differ from those defined within Windows 10/11. Windows 10/11 also allows graphical re-arranging of monitors.

Addressing one of the monitors is performed by its number. The monitor number can be set **in the file UR2\_cfg.dat** by the following entry (here set to 1):

Monitor#=1

After having defined the monitor number, it may happen that the UncertRadio window does not appear at its expected screen location. Then, testing different monitor numbers is required.

After the UncertRadio window has been built completely, the number of the monitor really being used can be displayed with the **Menu – Options – Monitor#**. For this query, the number of that monitor is obtained, in the rectangle of which the upper-left part of the UR window lies.

One of two versions before 2.3.08, a programmed re-positioning or re-sizing could occur. As the coordinates required for this operation were derived from the screen, one could observe in the case of more than one monitor, that the UR window was wider than the width of the actual monitor. In addition, such an oversized window could not be downsized.

**Note**: The application of **working with several monitors** within UncertRadio **is still in a test phase**.

**According to the present state, the following steps are recommended.**

There are three options available for the UR2\_cfg.dat file if **only one monitor** is connected to the PC:

a) apply the entry “Monitor#=0“, or   
b) omit this entry, or   
c) apply “Monitor#=1“ (in this case the separation of the MC graphics window is better separated from the UncertRadio window).

In the case of **more than one monitor** one should first try the options a) or b).

If the UncertRadio window does not appear on the desired monitor, the appropriate entry “Monitor#=” in UR2\_cfg.dat can only be found by trial.

In the present state, UncertRadio tries to find from Windows the appropriate coordinate values of the monitor **by the following algorithm**. The first step is to read the scaling, the parameter LogPixels, where LogPixels = 96 is defined as 100 %. UncertRadio has then a table of 20 possible width/height combinations of a monitor available. With the monitor coordinates read from Windows, it first tries to find a combination from the table exactly matching these coordinates without taking scaling into account. If this fails, in a second step the monitor coordinates are reduced by the scaling factor (dividing by (LogPixels/96)) and then selects from the 20 width/height combinations that one with the closest agreement.

## 1.4 Program version and application note

|  |  |
| --- | --- |
|  |  |

The Windows program UncertRadio is provided **free of charge**.

The program was developed by the author following state-of-the-art of science, standardization and technology. Nevertheless, NO GUARANTEE for the correctness of the user’s own results obtained with UncertRadio is given.

Information on possible errors in the program is highly appreciated!

**UncertRadio** is well suited for comparison calculations, e.g., for comparison with spreadsheet-based calculations.

## 1.5 UR Help and network drives

The Windows Help file UR2\_HELP\_EN.CHM, of the program cannot be used from a network drive because of Windows safety reasons. Therefore, it is recommended to install UR incl. its Windows Help on a local drive. However, it would be sufficient to move only the CHM Help file to a local drive. Within the configuration file UR2\_cfg.dat its full pathname can be defined by the entry Help\_path=.

The following may also be recommended:

* Open the context menu »Properties« by clicking on the CHM file with the RIGHT mouse button;
* open the tab »Safety«; therein, a text like » This file came from another computer and might be blocked to help protect this computer« is shown;
* click the button for allowing access.

# 2 Contents of the Program

## 2.1 Introduction

The program **UncertRadio** allows by inserting defining equations for **the evaluation of a measurement in the field of measuring activity** the **full calculation of the output quantity and its combined uncertainty (according to ISO GUM)** and of the **values of the Decision threshold and the Detection limit (according to DIN EN ISO 11929-1:2021, DIN EN ISO 11929-2:2021 and DIN EN ISO 11929-3:2021)**, which are closely related to uncertainty.

The program assumes the **ISO GUM interpretation which is based on the Bayesian theory of the measurement uncertainty (Bayesian statistics)**. The Bayesian measurement uncertainty does not have a statistical uncertainty, nor does it consider degrees of freedom, which makes it different from conventional (frequentist) statistics. Therefore, degrees of freedom need not to be treated in the program.

In the most measurement problems **measurements** of counts or of counting rates **are terminated if the counting time reaches its pre-set value**. For the case of pre-setting a count number, where the counting time is a random variable, section 7.20 shows how this is to be treated.

The present state of the program allows calculations **for up to three output quantities**.

The sequence of steps being treated is:

|  |  |
| --- | --- |
| 1. | input of a short text description of the measurement problem; |
| 2. | **input** of the **equations** defining the **output quantity** ***y*** of the measurement procedure; these define the **“evaluation model”**; the first of the equations must define the output quantity/ quantities;  Note: If more than one output quantity is to be treated, the calculation of the values of output quantity, uncertainty, uncertainty budget, Decision threshold and Detection limit, respectively, refers only to a single output quantity. By starting with a project, by default the first of the output quantities is “activated”; at a later stage another one can be selected in the main menu to be the “active one”; |
| 3. | automatic **extraction** of the **formula symbols** and **input of their meaning** (unit, meaning); automatic classification as independent and dependent symbols; manual addition of other symbols, not used explicitly in the equations; |
| 4. | selection of symbols which define the **net counting rate (*Rn*)** and the **gross counting rate**, only for the purpose of the calculation of Decision threshold and Detection limit; the net counting rate in this case must be the “**procedure dependent net counting rate**” in which the counting rate contribution due to interference from other radionuclides, usually obtained by calculation, is taken into account; |
| 5. | **input of measured values and their associated standard uncertainties** of input quantities (independent symbols) in table form; |
| 6. | for the **input of measurement uncertainties** their **associated distribution** can be chosen from a) **normal distribution**, b) **uniform distribution,** c) **triangular distribution** and d) **gamma distribution**, as well as a few others; half-widths of the latter two are converted by the program according to ISO GUM to standard uncertainties; for a complete list see [other distributions](#_6.11_Special_distributions).  Note: In the case of low-level applications with very low count numbers [**the so-called “(N+x) rule)“**](#URH_NP1REGEL_EN) (d) may be selected in order to improve the results, from which the associated counting rate variables are to be considered as being Gamma distributed. |
| 7. | for the input one can choose between **absolute and relative uncertainties;** |
| 8. | for counting rates or the number of counts **uncertainties can also be defined by formulae;** |
| 9. | the standard deviation of the gross count rate must be defined by a formula; this is the **“uncertainty function” (standard uncertainty) of the gross counting rate**, which is the basis for deriving values of Decision threshold and Detection limit; |
| 10. | **input of covariances between input quantities** which can be given as formulae or as values of correlation coefficients in tabular form; |
| 11. | numerical calculation of values and combined standard uncertainties for the quantities classified as dependent quantities and for the output quantity and of the **uncertainty budget**; consideration of covariances/correlations between input quantities is possible; calculation of the "[**best estimates and confidence limits based on a Bayesian method**](#URH_BestBayes_EN)" characterizing the value and standard uncertainty of the output quantity; |
| 12. | iterative numerical calculation of **Decision threshold and Detection limit** for the output quantity based on the numerical evaluation of its combined uncertainty taking covariances into account; |
| 13. | a [**Monte Carlo simulation**](#URH_MC_SIM_EN) can be started for an examination of the above-mentioned calculations which allows an independent calculation of the value of the output quantity and its combined uncertainty; partial derivations are not necessary in this case: for every input quantity classified as independent a value is drawn from its associated distribution from which a value of the output quantity is calculated using the defined equations. From the many-fold repetition of this step a statistical distribution is obtained for the output quantity which is used then to derive the “best estimate” form its mean and the combined uncertainty from its standard deviation; confidence limits are estimated as Quantiles of that distribution, whereas Decision threshold and Detection limits require to create separate MC distributions with modified values of the output quantity and their estimation by corresponding Quantiles; |
| 14. | finally, a **complete report can be created as a text file** containing all the equations, input values, uncertainty budget table and the final results - including those from the Mont Carlo simulation, the PDF file also contains the MC graphs. |
| 15. | The characteristic values obtained can be appended with “Save to CSV” as a single record to a CSV file. |

**UncertRadio is well suited for calculation comparison** for such solutions**,** which one may have already developed with spreadsheet calculations. The latter may get quite complex and sometimes are not so easily manageable especially with respect to the correct uncertainty propagation.

## 2.2 Numerical procedures

### 2.2.1 Structure of equations

The conditional equations defining the model for calculating the value of the **output quantity/quantities *y*** are written into a text field of the program. To improve the readability of equations auxiliary quantities may be inferred. The **formula symbols** are extracted automatically from the equations and are transferred into a table where they individually can be complemented by a **unit** and a **meaning**. After input of values and uncertainties of primarily measured quantities a “Function parser” is used for interpretation of the equations and calculation of the value and uncertainty of the output quantity ***y***.

For the process of editing the equations it is important to infer a quantity representing a **net counting rate,** called ***Rn*** in this help file, on which the output quantity depends linearly:

***y = Rn\*FL + FC*** (1)

Herein, the proportionality factor FL represents the procedure dependent calibration factor. FC considers further interference contributions, as e.g. one originating from the addition of a tracer activity before beginning with the radiochemical analysis.

The net counting rate *Rn* shall be understood as that net counting rate (more precise: procedure dependent net counting rate) from which all those contributions to the gross counting rate *Rg* which are not derived from the source contribution itself have been subtracted. The latter are not only the detector-related background *R0* but also blank contributions *Rbl* and, if applying a tracer solution (alpha spectrometry), additional blank contributions due to impurities in the tracer solution. Additionally, a calculated contribution *Rint* may be included due to interference by another radionuclide. **As an example, the procedure dependent net counting rate may then be:**

*Rn* = *Rg* – *R0* – *Rbl* – *Rint* . (2)

The constants *FL* and *FC* can be easily determined within the program for arbitrary types of equations, if these depend linearly on the net counting rate. This representation allows UncertRadio to solve Eq. (1) for a modified net counting rate value if the output quantity value were changed to y‘:

*Rn‘ = (y‘ – FC) / FL.* (3)

Similarly, the equation for a net counting rate *Rn* can be expressed more generally as a linear function of the gross count rate *Rg*:

*Rn = FB \* Rg - R0total*. (4)

In most cases the factor *FB* is equal to one; but FB may also differ from one. R0total is the sum of background contributions to be subtracted from the gross counting rate; see Eq. (2). At the beginning of computations, UncertRadio determines the values of *FB* and from this the fixed value R0total:

*R0total = FB\*Rb - Rn*. (5)

A net counting rate value Rn’ obtained by iterations within the detection limit calculations, is associated with a modified value of the gross counting rate:

*Rb‘ = (Rn‘ + R0total) / FB*. (6)

**Special feature**

All counting rates in equation (2) may also appear as to be multiplied with factors g, associated with uncertainties, as e.g.

*Rn* = *gb*\**Rb* – *g0*\**R0* – *gbl*\**Rbl* – *gint*\**Rint (7)*

**Non-linear dependence**

There may exist cases in which the dependence between output quantity and net counting rate, or, when using linear unfolding, between output quantity and the activity, is not linear. Consequently, the values *FC* und *FL* in Eq. (1) are only approximate ones and the inversion given by Eq. (3) is no longer correct.

Therefore, in addition to Eq. (1) and Eq. (3), two internal functions are used in UncertRadio:

* As an alternative to Eq. (1) a function **ActVal**(*Rn*) for calculating the value of the output quantity is used based on the function **RESULT** (s. below);
* For the reversion according to Eq. (3) a new function **RnetVal**(*y‘*) is used as an alternative; it uses a numerically working method according to Brent; it requires initial guess values for the lower and upper limit of the net counting rate values to be searched for, which are easily derived from the values of *y‘*, *FC* und *FL*.

### 2.2.2 Types of models

Regarding the relation between the output quantity and the net count rate, or the activity in the counting source in the case of linear unfolding, three types of models of measurement are considered in UncertRadio:

* positive linear model, including detection limit calculation
* negative linear model, including detection limit calculation
* only GUM, without detection limit calculation

The relation described in the section above, represents the most often encountered case of a **positive linear model**. In this case, the net count rate *Rn* increases with an increasing gross count rate. An activity is detected, if the gross count rate is significantly larger than the background count rate.

**Negative linear Model**

With the measurement model explained above, the net count rate Rn increases with increasing gross count rate; it is called positive linear. The activity is considered as detected if the gross count rate significantly exceeds the background count rate.

A **negative linear model** is characterized by a gross effect which must fall significantly below the “background” for the detection of the effect. In this case the difference in the expression ***Rn*** for the net effect is reversed:

***y = (R0 – Rb)\*FL + FC***  with Rb < R0 (8)

*Rb* and *R0* herein are not necessarily count rates. The project Rn-222-Emanation\_EN.txp is an example for it.

The case abbreviated as **„only GUM“** has only been introduced for the situation that only an uncertainty according to GUM is of interest for a measurement, but a detection limit shall not be calculated, as it occurs for instance in the case of weighing a sample.

### 2.2.3 Combined standard uncertainty of the output quantity

The combined standard uncertainty of the output quantity y is determined according to ISO GUM (ISO ***G***uide on ***U***ncertainty of ***M***easurement (1995); see also EURACHEM / CITAC Guide "Quantifying Uncertainty in Analytical Measurement" (2000)) using the “Gaussian law of propagating uncertainties” and taking covariances between individual input quantities.

For a type A standard uncertainty of a quantity which must be derived from a set of repeated measurements a necessary small statistical evaluation of mean and standard deviation has to be performed outside this program; only mean and standard deviation will be used in UncertRadio. Usually, with measurements of activities the greater part of uncertainties of quantities belongs to Type B. Within the framework of the **Bayesian theory of measurement** (Weise & Wöger, 1999; Weise et al., 2006) which is underlying the basics of this program an **explicit differentiation between quantities of type A and B is not necessary**. This is also the reason that degrees of freedom are not considered in this program.

The analytical derivation of formulae for the combined uncertainty for instance of a mass or volume dependent activity using the law of propagating uncertainties may easily yield a certain number of less or more complex formulae the correctness of which is often not easily being controlled. Therefore, a numerical procedure is applied.

The first step consists in transferring all quantities/parameters being required for the calculations, these may easily become more than 20, into a program array ***Messwert(i) = p(i)***. Then, in a subprogram RESULT the value of the output quantity is calculated from the values of the array elements *p(i)* by using the function parser.

Similarly, the known measurement uncertainties of the individual quantities/parameters are transferred to an array ***StdUnc(i) = u(i)***. For the calculation of the combined uncertainty a [**subroutine Uncpropa**](#URH_UNCPROPA_EN) is used to which the two arrays *p(i)* and *u(i)* are transferred. Covariances are considered in this subroutine. The sensitivity coefficients, i.e. the partial derivates of the function calculated with RESULT with respect to the array elements *p(i)*, are numerically approximated by differential quotients.

Further details: see [**Uncertainty propagation**](#URH_UncPropMeth_EN)

### 2.2.4 Iterative determination of Decision threshold and Detection limit

The **calculation of detection limits is based on DIN EN ISO 11929:2021** which is derived from Bayesian methods (see also Weise et al., 2006). It utilizes complete uncertainty propagation taking all individual uncertainties and covariances into account where the numerical calculations are based on the routines **RESULT** and **UncPropa**. The values of Decision threshold and Detection limit for the output quantity *y* are calculated by using an iterative procedure. In this procedure, the value of *y* is varied, now designated as “assumed value” . From this, **the iterated value of the gross counting rate** is obtained via calculating the net counting rate from . F**or each iteration step the combined standard uncertainty of , now called uncertainty function,** is in turn is derived from the easily calculated

The **Decision threshold** ***y\**** for the output quantity ***y*** is calculated according to ISO 11929 as follows:

, (7)

where is the normal quantile belonging to the error of first kind,  **UncPropa** is used for calculating the combined standard uncertainty of the output quantity under the constraint that the net counting rate is set equal to zero. This is easily done.

The **Detection limit** for the output quantity ***y*** is calculated as follows, where is the value of the Decision limit taking from the preceding step and is the normal quantile belonging to the error of second kind, 

. (8)

This represents an implicit equation for , because on the right-hand side of the equation the uncertainty *u* is to be calculated for a value of *Rn*, which corresponds to the value on the left-hand side; the latter is obtained by the inverse function which is easily established as *Rn = (y – FC)/FL* from the simple linear relationship between *y* and *Rn, y = FL\*Rn + FC.*

The solution of the implicit equation (8) is obtained by a simple iterative procedure which is demonstrated for the [**detection limit case**](#URH_NWGiterat_EN). The value of Factor is determined in the subprogram **RESULT**, while the uncertainty in Eq. (2) is calculated with the function subprogram **UncPropa**. In order to use UncPropa correctly, for each iteration step the corresponding value is obtained from the associated , which then is transformed to the gross counting rate, which in this example is stored in the array element . The uncertainty of the latter is calculated from the “uncertainty function (standard uncertainty) of the gross counting rate” which has been supplied to the program by the user. In the example the gross counting rate is calculated as if had been obtained by simple single-channel counting, which applies to most cases: .

The method by Brent is applied for the numerical iteration procedure.

**Special cases**

In the case of linear unfolding by using linear least squares analysis, e.g. in the evaluation of a decay curve, the fitting parameter for the desired net counting rate is that quantity which is varied by iteration for estimating Decision threshold and Detection limit (c.f. [**Note on Decision threshold and Detection limit with linear unfolding**](#URH_LSQ_NWG_EN)).

For determining the activity of a radionuclide from several gamma lines the quantity associated with this activity is the one of which the value is iterated (c.f. [**Method for calculating Decision threshold and Detection limit with Gamspk1**](#URH_GSPK1_NWG_EN)).

### 2.2.5 Preventing “hidden” covariances

For the following, it is assumed now that the arithmetic expression for the output quantity containing several expressions, e.g., , each of which being functions of input quantities . Often, the uncertainties are calculated first from which then is derived. If, however, there are some of the input quantities , contained in more than one of the expressions , then “hidden“ or “overlooked” covariances exist between some of the , which would have to be considered afterwards.

This problem does not occur during the uncertainty calculations within UncertRadio, because there the partial derivatives in its uncertainty propagation are always build from the equation of the output quantity. It is shown below, why this avoids the above problem.

The ansatz for the uncertainty propagation with partial derivatives which refer to the output quantity , is formulated as follows with the vector of input quantities:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |
|  |  | (2) |

At first, the square within the sum is evaluated:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Now, the summation over is performed for each of the six terms, while at the same time the partial derivatives of by are factored out of the sums:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Now, each individual sum over is representing a variance or a covariance of the expressions :

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Usually, a “hand-made“ uncertainty propagation by first applying a decomposition of into expressions or functions , only the first three terms in Eq. (6) are used, because covariances between the often are not expected; this may explain the term “hidden” covariances.

The result of Eq. (6) is just the one which has to be expected when “hidden“ covariances between the are explicitly taken into account. This demonstrates that these covariances are considered by UncertRadio, automatically, only because it uses within its uncertainty evaluation according to Eq. (1), partial derivatives directly of the output quantity.

### 2.2.6 Using switching variables in equations

The function parser (fparser) implemented in UncertRadio allows to apply **switching variables**, to which **only the two values 0 and 1 can be attributed**. Such a variable allows to **activate another variable b** with **b^1** or **to deactivate it by b^0**. In UncertRadio, these variables can be declared by attaching the string “\_Trigger“ to the symbol name. They are, therefore, also called “**trigger variables“**. Examples are: “min\_Trigger“, “kilo\_Trigger“; with “60^min\_Trigger“ or with “1000^kilo\_Trigger“ scaling factors of 60 (for minutes) or 1000 can be switched; see chapter 2.2.7. If a switching variable is to be used for count rate variables, it must contain the part “\_Trigger“ attached to e.g. „min“; then they can be identified by the program which in turn helps to prevent them from disturbing the process of finding such count rates which directly contribute to the net count rate (see chapter 2.3).

### 2.2.7 Calculation of physical units for dependent variables

UncertRadio contains a menu item which allows as a test to derive **the physical units of dependent variables** based on a numerical algorithm. The units of the input quantities, often given by “derived units” are changed to “basic units”; in addition to this, the associated scaling factors are determined. If, for example, a counting duration variable was given the unit “min“, an associated scaling factor of 60 is applied for changing to the basic unit “s”. The description of basic units and their derived units and of the algorithm for “calculating“ the units of dependent variables is given in [chapter 7.21](#URH_treatment_Units_EN).

A CSV file distributed with the program contains only a small number of units, which is nearly sufficient for measurements of radioactivity. If necessary, the basic units within the CSV file can be modified by the user.

By the menu item “Edit – test physical units“, the transformation to basic units can be tested. In UncertRadio’s text editor a comparison of “original“ and “converted“ units is shown for the list of quantities (symbols), as well as their associated values and uncertainties. At the begin of this list, a first error message is shown in case of conversion errors found by the program, which can indicate indirectly a wrong combination of units in the indicated number of the equation.

By programming, the routine behind this menu item was applied to all of UncertRadios example projects. There were indeed errors found, and it was necessary in most of the examples to replace the unit “1” of a detection probability by the unit “1/Bq/s“ in order to get the unit “Bq” for an activity as output quantity. More details are given in [chapter 7.21.3](#URH_test_Units_EN).

## Equations as tree topology

The equations for calculating the value of the output quantity, as being set up in section 2.2.1, are **hierarchical equations**. They form a list of dependent quantities (of number *nab*), followed by the list of independent input quantities (of number *nmu*). Therefore, numbers from 1 to (*nab*+*nmu*) are attributed to the quantities (also called symbols). This list may be considered as if it were a ladder with (*nab*+*nmu*) steps, or a decay scheme of a decaying atomic nucleus; on each step (or level) a quantity symbol resides. According to their associated auxiliary equations, the dependent quantities are related with other symbols on lower positions of the ladder (or level). Connecting these symbols by lines generates a tree structure, which is comparable to a series of allowed level transitions of a decaying atomic nucleus. The series of connecting lines down to the ladder step of an independent input quantity can be compared with a cascade of level transitions of a nucleus ending at the ground state level.

The following is restricted to applications not using linear unfolding.

Such “symbol cascades“ can be generated from all „transitions“ (*i*, *j*)=*i*🡪*j* between symbols *i* and *j* within a cascade. They can be found by using a recursive numerical algorithm.

This method is especially applied to find out for a net count rate *Rn*, being proportional to the output quantity value, on which individual count rate contributions *Ri* it depends. Furthermore, the count rates *Ri* have two additional properties:

1. a square-root-based uncertainty function like sqrt(*Ri*/*ti*) can be attributed to them, or,
2. they may be based on count numbers *Ni*, also being associated with uncertainty formulae like (sqrt(*Ni*)), or for which special distribution types are declared, e.g., the gamma distribution (“x+1”) or a Poisson/binomial distribution.

With taking these additional properties into account (called „rules“ below), in most cases those symbols can be identified, which represent a count number, including also the associated counting duration. Then, by going one step back within the affected hierarchy ladder, the symbol representing the associated count rate *Ri*=*Ni* / *ti* is found.

Knowing the relation between the gross count rate *Rg*, the gross count number *Ng* and the counting duration *tg*, and their symbol numbers within a cascade, allows, for deriving decision threshold and detection limit, to generate a modification from *Rg* to *Rg*~ from the related modification from *Ng* to *Ng*~. For simplification of this step, index fields are generated within UncertRadio which point from a count rate to the number of counts and to the counting duration, and vice versa. This, however, requires that not only count rates *Ri* alone are defined in the equations, but also the equations *Ri=Ni* / ti. This results in the recommendation, to follow this in working with UncertRadio.

**Example** Ra226\_U235-at-186keV\_EN.txp.

Equations (*nab*=8, *nmu*=10):

Formeltext=

1 : cRa = Phi \* RRa

2 : Phi = 1. / (eps \* pRA \* mp)

3 : RRa = RS - RU5

4 : RS = Rb - RT - RnNE

5 : RU5 = AU5 \* Ufakt

6 : Ufakt = eps \* pU5 \* mp

7 : Rg = Ng / tm

8 : RT = NT / tm



Table of transitions *i* 🡪 *j*:

ndep eqnum(ndep) synum(ndep) opnum(ndep) Symb(i) Symb(j)

=i =j

---------------------------------------------------------------

1 3 4 - RRa RS

2 3 5 RRa RU5

3 4 7 - RS Rb

4 4 8 - RS RT

5 4 12 RS RnNE

6 5 13 \* RU5 AU5

7 5 6 RU5 Ufakt

8 6 9 \* Ufakt eps

9 6 14 \* Ufakt pU5

10 6 11 Ufakt mp

11 7 15 / Rb Nb

12 7 16 Rb tm

13 8 17 / RT NT

14 8 16 RT tm

Table of cascades (chain) and three identified count rates as part of the net count rate:

nc i j kcnt ktime krate rule Symbol chain

-----------------------------------------------------------------------

1 7 15 15 15 7 A5 Rg 3 4 7 15

2 7 16 0 0 0 3 4 7 16

3 8 17 17 17 8 A3 RT 3 4 8 17

4 8 16 0 0 0 3 4 8 16

5 4 12 0 0 12 A6 RnNE 3 4 12

6 5 13 0 0 0 3 5 13

7 6 9 0 0 0 3 5 6 9

8 6 14 0 0 0 3 5 6 14

9 6 11 0 0 0 3 5 6 11

Table of index fields of counting duration (iptr\_time) and number of counts (iptr\_cnt) to the count rate (iptr\_rate)

(*RnNE* is defined only as a net count rate of the background measurement)

i iptr\_time iptr\_cnt iptr\_rate Symbol

--------------------------------------------

7 16 15 7 Rb

8 16 17 8 RT

12 0 0 0 RnNE

Among the example projects belonging to UncertRadio are two, for which the algorithm shortly introduced above in fact finds specific count rate symbols two times:

BSH\_total-gamma\_var2\_EN.txp

DWD\_sr89\_sr90\_TDCR\_procedure\_EN.txp

In the first one, this result leads to the conclusion, that the equations constituting the net count rate, have not been simplified enough. In fact, it can be demonstrated that the corresponding equations can be re-worked algebraically such that the equations of the alternative example, BSH\_total-gamma\_var1\_EN.txp, are exactly met.

In the second example mentioned above, the equation for *Rn\_s* for calculation the Sr-90 activity is rather complex, so that both, *R0\_s* and *R0\_c*, appear twice in them, also in a non-linear form.

**Note**: While running the QC batch mode processing an additional file fort.64 is produced showing in short form the identified count rate contributions to the net count rate (projects not using linear unfolding). *Meanwhile, this option is deactivated*.

**Note**: The example given above shows that the gross count rate Rg is the first in the list of count rates contributing to the net count rate. This characteristic can be used for the internal checking whether the correct gross count rate symbol has been selected within the TAB „Equations“, because **the gross count rate is always the first of the count rates in the expression for the net count rate.**

## 2.4 Applied programming methods

The program was developed under Windows in Fortran 90/95/2008.

The Fortran code uses “character arrays of deferred shape” by creating a special user-type for them. This means, different elements of a character array can have different sizes. This required to use extensively memory allocation/deallocation of variables, also for integer or real arrays. This also allows to define symbols in UncertRadio equations with more than 20 characters.

The Fortran code consists of about 430 routines / functions, with about 47 thousand lines (without comment and blank lines, without GTK-Fortran and PLPLOT routines).

The following programming tools were used for preparing the program with a Windows-GUI:

* ***GNUFortran***; 64 bit, Version 14.1.0
* ***GTK3+***, version 3.24.41, a Graphical User Interface (GUI) developed for C users (<https://www.gtk.org>);
* ***GTK-Fortran,*** Version 20.04 from 2020-05-07, a Fortran interface to the C functions of GTK+ (<https://github.com/jerryd/gtk-fortran/wiki>);
* ***Glade Interface Designer*** for Windows (version 3.40.0) for graphically setting up the GUI (<http://glade.gnome.org>);
* ***PLplot*** *5.15.0* for graphical plotting (<http://plplot.sourceforge.net/> ).

The tools GNUFortran-Compiler, GTK+3 and Glade were applied as packages from an installation of **MSYS2** (<https://www.msys2.org/wiki/MSYS2-introduction/>). The actual CHM version of the UR Windows Help was created with the **NüHelp** tool (version v2018.04.23; <https://sourceforge.net/projects/nuhelp/> ).

For obtaining numerical results from the equations defined by the user in a text field a Public Domain ***Function Parser Module*** was used which was made available by Roland Schmehl (University of Karlsruhe) as Fortran 90 source on a website which does no longer exist.

A generator for Gamma-distributed random numbers according to Marsaglia and Wang (2000) was used in the form which is available on Alan Miller’s famous website for Fortran90 routines.

## 2.5 Internal numerical precision

The program previously worked internally with “extended precision“ (real(10)), by which the precision of numbers is better by about three decimals. This has the advantage that especially numerical partial derivatives can be calculated with about three more precise decimals.

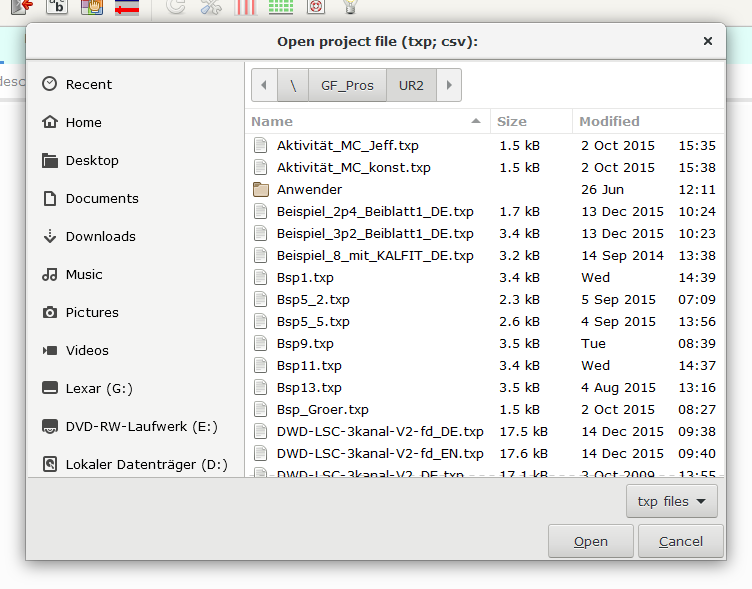
This does not mean, however, that the value of the detection limit has the same precision. Its precision is limited by the convergence criterion (1.0E‑06 \* detection limit) of the associated iteration loop, which is chosen as a compromise between precision and required computation time.

Numbers written as integers by the user in equations are internally treated as (real(10)) values. Example: ln(2), which is used to define decay constants, is correctly interpreted as real(10) when given as “log(2)” or “log(2.)” by the user.

## 2.6 Additional Notes

***File Selection Dialog***

Compared to Windows, the graphical layout of this dialog has changed. This originates in using platform-independent programming tools.



The way of scrolling with the scroll bar at the right margin has changed. The scroll bar becomes visible when the mouse cursor approaches the right dialog margin. The context menu of the right mouse button opens further layout options.

Note: If this dialog is used in the mode “**save as**“, *the desired file extension of the file name must explicitly be given or edited in the name field (at the top of the dialog).* Only the pure filename has to be inserted into this field, the desired path name is selected in the dialog elements below.

Clicking on “**Recently used**“ shows a list of recently used files. The filenames are hold by the *RecentManager* of GTK; the latter works with a file “recently-used.xbel“, which e.g. for WINDOWS is found in the folder “c:\users\user\AppData\Local\“ (the actual Windows name of the user replaces “user“ in the folder name).

***Input to tables***

The input of a value into a cell of a table has to be finalized with the enter key.

***Column blocks in tables***

Column blocks can no longer be selected with UR tables, i.e. the export of such blocks to e.g. Excel is not possible. However, the reverse way is possible: the import of a column block taken from e.g. an Excel file, or from the text editor Notepad ++, into a column block of equal size in an UR table; [see also](#URH_TABTRICKS_EN).

For selecting a whole row click into the right part of a cell in this row.

## 2.7 Literature

AKU, 2008*. Moderne Routine- und Schnellmethoden zur Bestimmung von Sr-89 und Sr-90 bei der Umweltüberwachung*. Bericht einer Ad-hoc-Arbeitsgruppe des Arbeitskreises Umweltüber­wachung (AKU). Bericht FS-08-147-AKU des Fachverbandes für Strahlenschutz. TÜV Media GmbH, Köln.

Barlow, R.J., 1999: *Statistics. A Guide to the Use of Statistical Methods in the Physical Sciences.* The Manchester Physics Series. John Wiley & Sons Ltd., Chichester, New York, 204 S.

Blobel, V., Lohrmann, E., 1998. *Statistische und numerische Methoden der Datenanalyse*. B.G. Teubner Stuttgart-Leipzig, 358 S.

Brandt, S., 1999: *Datenanalyse. Mit statistischen Methoden und Computerprogrammen*; 4. Auflage. Spektrum, Akademischer Verlag, Heidelberg-Berlin, 646 S.

Cox, M.G., Harris, P.M., 2001: *Measurement Uncertainty and the Propagation of distributions*. NPL, UK, Paper presented at the 10th International Metrology Congress, Saint-Louis, France, 22-25th October 2001.

Cox, M.G., Forbes, A.B., Harris, P.M., Smith, I.M., 2004: The classification and solution of regression problems for calibration*.* NPL Report CMSC 24/03, (chapter 6.3), National Physics Laboratory, Teddington, UK, 46

http://www.npl.co.uk/ssfm/download/nplreports.html.

Cox, M., Harris, P., Nam, G., Thomas, D., 2006: *The Use of a Monte Carlo Method for Uncertainty Calculation, with an Application to the Measurement of Neutron Ambient Dose Equivalent Rate*. Radiation Protection Dosimetry 121, pp. 12-23.

Cox, M.G., Eiø, C., Mana, G. and Pennecchi, F, 2006b. *The generalized weighted mean of correlated quantities*. Metrologia 43 S268-S275

EURACHEM/CITAC, Guide CG 4, 2000. *Quantifying uncertainty in analytical measurement*. Second edn., 120 S. <http://www.eurachem.ul.pt/guides/QUAM2000-1.pdf>.

Gavin, H. P., 2022. *The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems*. Department of Civil and Environmental Engineering, Duke University

http://www.duke.edu/∼hpgavin/m-files/lm.m

Gilmore, G.: Practical Gamma-Ray Spectrometry. 2nd Edition; J. Wiley & Sons Ltd; 2008.

Hauschild, T., Jentschel, M., 2001. *Comparison of maximum likelihood estimation and chi-square statistics applied to counting experiments*. Nucl. Instr. & Meth A 457 (1-2), S 384-401.

Hoover, W. E., 1984: *Algorithms For Confidence Circles and Ellipses*. NOAA Technical Report NOS 107 C&GS 3; Charting and Geodetic Services; Rockville, MD; September 1984

<http://www.ngs.noaa.gov/PUBS_LIB/AlgorithmsForConfidenceCirclesAndEllipses_TR_NOS107_CGS3.pdf>

International Organisation for Standardisation, 1993. *Guide to the Expression of Uncertainty in Measurement (GUM)*. (Geneva: ISO), corrected reprint (1995), also as ENV 13005 (1999).

International Organisation for Standardisation, 2010. Determination of the characteristic limits (decision threshold, detection limit and limits of the confidence interval) for measurements of ionizing radiation — Fundamentals and application. (Geneva: ISO), 2010.

International Organisation for Standardisation, 2019. Determination of the characteristic limits (decision threshold, detection limit and limits of the coverage interval) for measurements of ionizing radiation — Fundamentals and application. Part 1: Elementary applications (Geneva: ISO), 2019.

International Organisation for Standardisation, 2019. Determination of the characteristic limits (decision threshold, detection limit and limits of the coverage interval) for measurements of ionizing radiation — Fundamentals and application. Part 2: Advanced applications (Geneva: ISO), 2019.

International Organisation for Standardisation, 2019. Determination of the characteristic limits (decision threshold, detection limit and limits of the coverage interval) for measurements of ionizing radiation — Fundamentals and application. Part 3: Applications to unfolding methods (Geneva: ISO), 2019.

International Safety Research, Safety Support Series, 2013. Radiation Counting Statistics. Volume 1. Canada.

Janßen, H., 2004. *Determination of Strontium-89 and Strontium-90 in soils and sediments.* In: Quantifying uncertainty in nuclear analytical measurements, IAEA-TECDOC-1401, pp. 149-166.

JCGM 101:2008. *Evaluation of measurement data — Supplement 1 to the “Guide to the expression of uncertainty in measurement” — Propagation of distributions using a Monte Carlo method* GUM, Joint Committee for Guides in Metrology, 2008. (GUM Supplement 1)

JCGM 102:2011. *Evaluation of measurement data – Supplement 2 to the "Guide to the expression of uncertainty in measurement" – Extension to any number of output quantities*. Joint Committee for Guides in Metrology; 2011; http://www.bipm.org/en/publications/guides/gum.htm

Kacker, R.N., Datla, R.U., Parr, A.C, 2002. *Combined result and associated uncertainty from interlaboratory evaluations based on the ISO Guide.* Metrologia 39, 279-293.

Kacker, R.N., Datla, R.U., Parr, A.C., 2004. *Statistical analysis of CIPM key comparisons based on the ISO Guide*. Metrologia 41, 340-352.

Kanisch, G., 2004. *Quantifying Uncertainties in the Alpha-spectrometric Analysis of Environmental Samples*. In: Quantifying uncertainty in nuclear analytical measurements, IAEA-TECDOC 1401, Vienna, pp. 127-139;

Kanisch, G., 2016. *Generalized evaluation of environmental radioactivity measurements with UncertRadio. Part I: Methods with linear unfolding*. Applied radiation and Isotopes 110, 28–41; [doi:10.1016/j.apradiso.2015.12.003](http://dx.doi.org/10.1016/j.apradiso.2015.12.003)

Kanisch, G., 2016. *Generalized evaluation of environmental radioactivity measurements with UncertRadio. Part II: Methods without linear unfolding*. Applied radiation and Isotopes 110, 74-86;

http://dx.doi.org/10.1016/j.apradiso.2015.12.003

Kessel, R., Kacker, R., Berglund, M., 2006. *Coefficient of contribution to the combined standard uncertainty*. Metrologia 43, S189-S195.

Knoll, G.F.. *Radiation Detection and Measurement*, 2nd edition, John Wiley, NewYork,1989, pp. 96-99

Laurence, T. A., Chromy, B., 2009. *Efficient Levenberg-Marquardt Minimization of the Maximum Likelihood Estimator for Poisson Deviates*. Report LLNL-JRNL-420247, November 13, 2009.

Marsaglia, G., Tsang, W.W., 2000. *A Simple Method for Generating Gamma Variables*. ACM Transactions on Mathematical Software, Vol. 26, No. 3, 363–372.

Mathews, I.P., Kouris, K., Jones, M.C., Spyrou, N.M.: Theoretical and experimental investigations on the applicability of the Poisson and Ruark-DeVol statistical density functions in the theory of radioactive decay and counting. Nucl. Instr. Meth. 171 (1979), 369-375.

Michel, R., 2000: *Quality assurance of nuclear analytical techniques based on Bayesian characteristic limits*. J. Radioanal. and Nucl. Chem. 245: 137-144.

Michel, R., Kirchhoff, K., 1999. *Nachweis-, Erkennungs- und Vertrauensgrenzen bei Kernstrahlungsmessungen*. Fachverband für Strahlenschutz e.V., Köln: TÜV-Verlag, Publikation FS-99-108-AKSIGMA, ISSN 1013-4506, 157 S.

Miller, A. Alan Miller’s Fortran Software. <https://jblevins.org/mirror/amiller/>

Moreno, J., Vajda, N., Burns, K., Danesi, P.R., De Regge, P., A. Fajgelj, A., 2004. *Radiochemical determination of Strontium-90 in environmental samples by Liquid Scintillation Counting*. In: Quantifying uncertainty in nuclear analytical measurements, IAEA-TECDOC-1401, pp. 167-193.

Pengra, D., 2008: *Counting statistics of random events: A tutorial*. 9 S.

http://courses.washington.edu/phys433/muon\_counting/counting\_stats\_tutorial\_b.pdf

Pishro-Nik, H., *Introduction to Probability*:

<https://www.probabilitycourse.com/chapter11/11_1_2_basic_concepts_of_the_poisson_process.php>

Pommé, S., Keightley, J., 2007. *Countrate estimation of a Poisson process: unbiased fit versus central moment analysis of time interval spectra*. Applied Modeling and Computations in Nuclear Science. In: Semkow, T.M., Pommé, S., Jerome, S.M., Strom, D.J. (Eds.), ACS Symposium Series 945. American Chemical Society, Washington, DC, pp.316–334.2007.ISBN0-8412-3982-7.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., 1992. *Numerical Recipes in FORTRAN*, second edn. Cambridge: Cambridge Unversity Press.

Ratel G., Michotte, C., Bochud, F. O.: Uncertainty of combined activity estimations. Metrologia 52 (2015) S30–S41.

Rusconi, R., Forte, M., Caresana, M., Bellinzona, S., Cazzaniga, M.T., Sgorbati, G., 2006. *The evaluation of uncertainty in low-level LSC measurements of water samples.* Appl. Radiat. Isot. 64, 1124-1129.

Salma, I., Zemplén-Papp, É.: Experimental investigation of statistical models describing distribution of counts. Nucl. Instr. Meth. A 312 (1992), 591-597.

Semkow, T.M.: Bayesian Inference from the Binomial and Poisson Process for Multiple Sampling. In: T.M. Semkow, S. Pommé, S.M. Jerome, D.L. Strom (Ed.): Applied Modeling and Computations in Nuclear Science. ACS Symposium Series 945, ACS, Oxford University Press, 2007.

Spyrou, N.M., Foster, J., Jones, M.C., Kouris, K., Matthews, I.P.: Should the Poisson statistical density function be used in the measurement of short-lived isotopes? J. Radioanal. Chem. 61 (1981), 121-130.

Thompson, M.A., 2015: *Gaussian Statistics Lecture.*

<http://www.hep.phy.cam.ac.uk/~thomson/lectures/statistics/GaussianStatistics_Handout.pdf>

Weise, K., Hübel, K., Michel, R., Rose, E., Schläger, M., Schrammel, D., Täschner, M., 2004. *Nachweisgrenze und Erkennungsgrenze bei Kernstrahlungsmessungen: Spezielle Anwendungen. Vorschlag für eine Norm.* Fachverband für Strahlenschutz e.V., Köln: TÜV-Verlag, Publikation FS-04-127-AKSIGMA, ISSN 1013-4506, 31 S.

Weise, K., Wöger, W., 1999. *Meßunsicherheit und Meßdatenauswertung*. Verlag Wiley-VCH Weinheim, 345 S.

Weise, K., Hübel, K., Rose, E., Schläger, M., Schrammel, D. Täschner, M., Michel, R., 2006. *Bayesian decision threshold, detection limit and confidence limits in ionizing-radiation measurement*. Radiat. Prot. Dosimetry 121(1), 52 – 63.

Weise, K., Kanisch, G., Michel, R., Schläger, M., Schrammel, D., Täschner, M., 2009. *Monte Carlo determination of the characteristic limits in measurements of ionizing radiation – Fundamentals and numerics*. Radiation Protection Dosimetry 135 (3), 169–196.

Weise, K., Kanisch, G., Michel, R., Schläger, M., Schrammel, D., Täschner, M., 2013. *Characteristic values in measurements of ionizing radiation – Materials for a critical discussion on Fundamentals and alternatives.* Fachverband für Strahlenschutz e.V., Köln: TÜV-Verlag, Publikation FS-2013-167-AKSIGMA, ISSN 1013-4506, 51 pp.

Wübbeler, G., Krystek, M., Elster, C., 2008. *Evaluation of measurement uncertainty and its numerical calculation by a Monte Carlo method.* Meas. Sci. Technol. 19, 084009 (4pp)

# 3 First steps

## 3.1 Starting the program and the main menu

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Menu File**

After having started the program, it is ready for dealing with a new measurement evaluation which is called **project**. The TABs “Procedure” and “Equations” are enabled.

A measurement problem which is already existing as a project file (extension .txp) can be loaded into the program under the menu item “**File – Load Project**” or with the icon document-open.png, which is automatically followed by the complete sequence of calculations which is finished when the TAB “Results” is enabled and made active. This may take some seconds which is pointed out also by an additional dialog, which vanishes when all calculations are done. Now, the user may work on that project.

If problems occur during the automatic sequence of calculations through the TABs while the project is loading, this sequence can be omitted with the menu item “**Options – Project Load – without calculations**“.

Under the menu item “**File - Save Project**” or with the icon document-save.pnga measurement evaluation being in progress can be saved as project file (extension .txp) under the same file name or it may be saved under a different project file name with the menu item “**File - Save Project As**” or with document-save-as.png.With “**File - Close Project**” or with  the project file can be closed. A csv file format can also be selected for loading or saving a project file.

**Menu Edit**

A report file “Report.txt” describing the present status of the project can be produced under the menu item “**Edit - Report**”. The contents of this file may be displayed with the internal text editor via the TAB “Text Editor” or saved with another filename. The results of all output quantities are written to the report file, starting from the uncertainty budgets for the second or third output quantity.

If more than one output quantities are involved, then, under the menu item "**Edit - Select output quantity**", one of these can be selected and to which then the calculations of its uncertainty, the uncertainty budget, Detection threshold and Detection limit refer to.

Note: If **another output quantity is selected** by the user, this implies that the **selections of the gross and net counting rate symbols must be changed accordingly**, unless the evaluation method is linear unfolding. The program then switches to the TAB “Equations” and gives an appropriate hint in the rightmost field of the status bar. If it is, however, a method of linear unfolding, where net and gross counting rates need not to be selected manually, all steps of calculations up to the TAB “Results” are performed automatically in one step.

The menu item “**Edit – Decay curve**“ allows editing some sub-dialogs and the primary fitting results if the procedure of linear unfolding was invoked by a call to **LINFIT(..)** within the equations:

* Sub-menu “**model of decay curve**“, or, equivalently the icon  in the toolbar:

allows editing parameters of the evaluation model;

* Sub-menu “**data input**“, or, equivalently the icon  in the toolbar:

invokes the sub-dialog for editing the input data of the decay curve;

* Sub-menu “**Curve-fit table**“, or, equivalently the icon  in the toolbar:

opens an editor window for viewing the primary fitting results.

If parameters or data have been modified while working within these sub-dialogs the evaluation is re-started and terminated at the TAB “**Results**“.

The menu item “**Edit – Gamma spect**“ allows editing of single dialogs or result, if the linear unfolding was activated by a call to **Gamspk1(..)** within the equations.

* Sub-menu “**Edit gamma lines**“, or, equivalently, the icon  within the toolbar:

This calls the dialog for editing the individual gamma line data;

* Sub-menu “**Average line activities**“, or, equivalently, the icon  within the toolbar:

This opens the Editor window for inspection of the results obtained for the weighted mean.

If changes have occurred within these sub-menus, the evaluation is repeated through to the TAB “Results“.

The menu item “**Edit – Calibration curve”** invokes a dialog, which allows the data input of a calibration curve, to fit a polynomial to it and to take for a specific calibration point value and uncertainty from the latter, which in turn are used in UR then.

The new menu item “**Edit – Change symbol name**“ allows to change the name of a specific symbol throughout the dialogs and program internal fields or arrays. The specification and unit associated with this symbol are maintained. If a symbol name needs to be changed, this should be done with this menu item instead of changing this name directly in the equations.

The new menu item “**Edit – Serial evaluation**“ has been introduced allowing the manifold evaluation of a project with partially modified input quantity values/uncertainties. The description of this new option is given in the new section 5.6 in chapter 5.

The new menu item “**Edit – Batch evaluation of projects**“ has been introduced allowing the serial evaluation of a a number of different project files .The description of this new option is given in the new section 5.7 in chapter 5.

The four parameters of the case of a binomial+Poisson distributed count number can be edited or re-edited using the dialog which is loaded under the menu item “**Edit – Set binomial/poisson case**“.

Under the menu item “**Edit – test physical units**” the consistency of the physical units of the quantities can be checked by an numerical algorithm. This test is described in detail in section 7.21.

If a project makes use of calculating the time dependent behaviour of a radioactive decay chain with two or more member, the menu item **“Edit – Edit decay chain”** allows to select a decay chain from a file with few pre-defined decay chains and to edit some other measurement-related conditions. A detailed description is given in section 6.14.

It may happen that, due to an error having occurred, the project can no longer be opened, it requires to be set up again. Instead of a completely new input of all the values and uncertainties of input quantities, it would then be helpful to take such values from the defect file variant of the project, if these still exist. This step is supported by using the menu item **Edit – Load missing values from project variant.** This requires only the input of the filename of the project variant. Values and uncertainties still existing in the project variant are transferred into the actually opened project for symbols with equal names.

**Menu Options**

With the menu item **“Options – pre-settings”** the values of, e.g., the two Quantiles of the normal distribution can be defined corresponding to the probabilities  and of the errors of first and second kind. The **Language** can also be selected there, with which UR shall be used (German or English).

Under the menu item “**Options – Project Load**” one may choose whether the calculations during the automatic run through the TABs while the project is loaded shall be switched off; at program start “with calculations” is activated.

If using linear unfolding (invoked by a call to Linfit(), the menu item „**Options – LSQ-Export to R**“ exports some input data of the fitting routine into some extra text files, e.g. UR-Export-to-R.txt, which can be used for import into the statistics package **R** which allows a comparison of the results between UR and **R**. By default, this option is activated.

The menu item “**Options – Model type“** allows to select from three different types of measurement models:

* **positive linear, with detection limit**:

the output value ***increases*** linearly with increasing “gross” quantity (this is the model type already used for evaluating activity and dosimetric measurements);

* **GUM only, without detection limit**:

only value and uncertainty are to be determined, e.g., determining a mass by weighing. In this model type neither a gross count rate nor a net count rate nor a detection limit are required.

* **negative linear, with detection limit (new)**:

the output value ***decreases*** linearly with increasing “gross” quantity.

The case of determining the detection limit for the emanation coefficient of Rn-222 may serve as an example, where in the Ra-226 source the (non-emanated) Rn-222 (Bi-214/Pb-214) activity must become smaller than that of Ra-226 in order to detect the emanation (both activities are measured in the Ra-226 source by gamma-ray spectrometry).

For a project using linear unfolding with more than two output quantities a confidence ellipse may displayed graphically for each pair of quantities under the menu item “**Options – Calculate confidence ellipse”**. The correlation matrix is also shown in the associated dialog.

The menu item **QC-Batch-Test** is described in section 3.9.

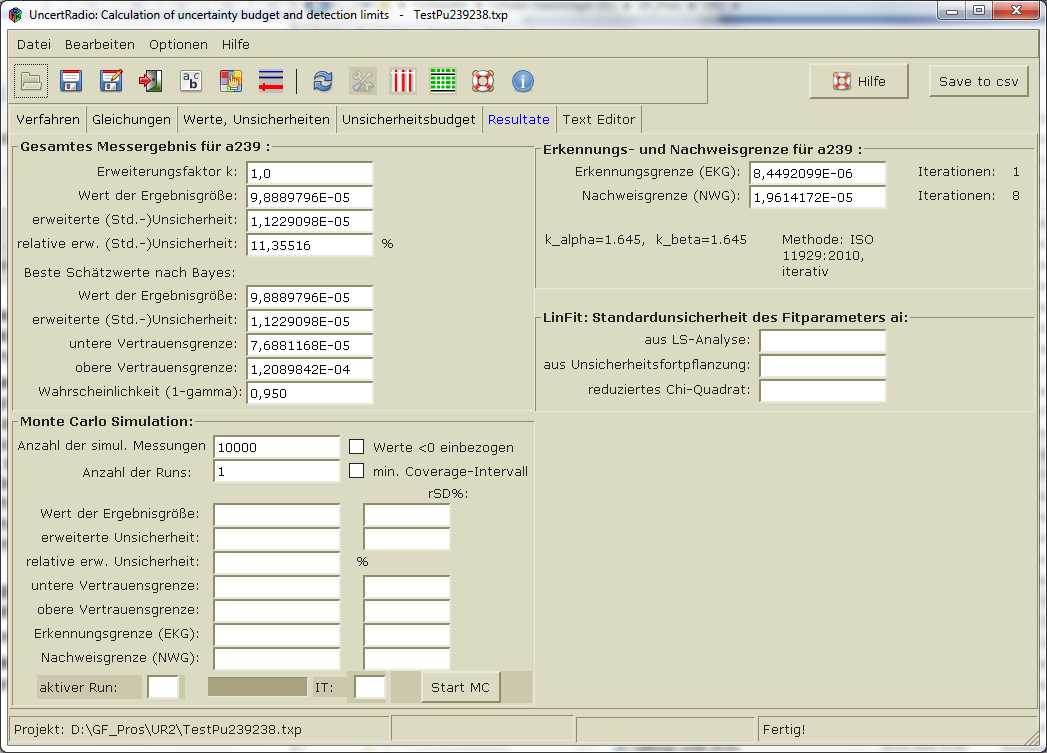
The menu item **Monitor#** allows to obtain the number of the monitor holding the UncertRadio window.

From the **remaining icons in the toolbar** the more important ones are:

* the **“update icon“**view-refresh.png, by which the calculations from the TAB “Values, Uncertainties“ through the TAB “Results“ can be performed in a single step, after changes in e.g. input data were observed;
* the “**delete rows icon“ ** allows to remove such rows which have been selected in advance by the mouse within grids, such as “Table of Symbols“ and others, also in other dialogs;

a block of rows may also be selected for this purpose:

select the upper row by mouse click, hold the shift key pressed down and click into the lower row;

* the CHM Help can be invoked with the icon help-contents.png;
* a page of the CHM Help for advices in case of problems can be invoked with the icon dialog-information.png;
* the “**fontname icon**“  allows choosing fontname and fontsize;
* background-colors can be modified by the “**color icon**“ .

*--> At present, changed colors* ***cannot*** *be transferred into the program window.*

* the “**mean-handling icon**“  allows input of values of a variable and the selection of such variable and of the type of mean.
* The icon  invokes a dialog showing the actual **parameters of a special distribution density** connected to an input quantity. This requires that the row of this input quantity within the table “values, uncertainties“ is highlighted.
* Short informations about special UR functions can be displayed by the icon Ein Bild, das Text enthält.

  Automatisch generierte Beschreibung.

**User guidance is given in the status bar at the bottom of the UncertRadio window, in the right-most field. If a project has been changed in some details this is indicated in the status bar to the left of the latter (“unsaved”).**

For working with tables: see [**edit tables**](#URH_TABTRICKS_EN)**.**

A certain number of projects files have been added to the program showing different examples of measurement evaluation. For an introduction it is recommended to load such an example project and go through it: [**“How to view stepwise an already existing project”**](#URH_PRANSICHT_EN).

## 3.2 Viewing an existing project

To become acquainted with the functioning of the program it is recommended to look into existing example projects. The sequence of steps and the buttons and tabs to be clicked as shown below is nearly the same when creating a new project.

**User guidance is presented in the last field of the status bar at the bottom of the UncertRadio window.**

 File - Load Project document-open.png : Load an existing project file (extensions txp or csv) of a measurement problem into the program, whereby all calculations are done and, finally, the TAB “Results” becomes visible.

 select TAB “Procedure” for comments on that procedure

 select TAB “Equations“

 click button “Load symbols from equations“

 click button “Load symbols from finalized symbol table“

* click button “Accept all“ -> this enables the TAB “Values, Uncertainties“

 look at the equations and the table of symbols

 select TAB “Values, uncertainties“

 click Button “Calculation of uncertainties“ -> this enables the TAB “Uncertainty budget“

 look at details in the uncertainty table

 select TAB “Uncertainty budget” showing the uncertainty budget of the output quantity

 select TAB “Results” showing all results including Decision threshold and Detection limit

 a Monte Carlo simulation of the calculations as an alternative method may be started by clicking the button TAB “Start“

 Closing the project file , loading the next project (see above), or terminating the program: File – Quit program.

**Note regarding safety:**

If one has changed values while going only through the different parts of the program, one should NOT save the project when the program asks for it, because otherwise the project file would be modified.

Scrolling within tables or in the equations field is possible with the mouse wheel after clicking within that table of field.

## 3.3 Examples for trial

### 3.3.1 List of example projects

Note: In most cases the associated literature for an example project is cited within the projects; see TAB “Procedure” after loading a project. See also Help topic “Used literature”.

In the mid of the year 2021, all projects were checked with respect to the correct use of physical units. Only very few errors were found and corrected. See chapter [2.27](#URH_preview_phys_Units_EN) for the necessary changes.

|  |
| --- |
| All example project files indicated by background color can be considered as being evaluated, because they originate from associated publications, or they were confirmed by colleagues through independent intercomparison calculations. |

|  |  |
| --- | --- |
| **Examples with 1 output quantity:** | |
| ***without linear unfolding:*** |  |
| ISO-Example-1a\_EN.txp | Alpha activity concentration in liquid material: measurement with an Alpha detector; corresponds to example 1a, section D.2.1, ISO 11929:2010 |
| ISO-Example-1b\_EN.txp | Alpha activity concentration in liquid material: measurement with a rate meter; corresponds to Example 1b, Section D.2.2, ISO 11929:2010 |
| ISO-Example-2a\_EN.txp | Sr-90 in soil, several determinations; corresponds to Example 2, Section D.3.1, ISO 11929:2010; DL 5 % too small |
| ISO-Example-2a\_V2\_EN.txp | simplified version of ISO-Example-2a\_EN.txp |
| ISO-Example-2b\_EN.txp | Sr-90 in soil, several determinations; corresponds to Example 2, Section D.3.2, ISO 11929:2010; |
| ISO-Example-2b\_V2\_EN.txp | simplified version of ISO-Example-2b\_EN.txp |
| ISO-Example-3a\_EN.txp | I-131 accumulation on air filter; corresponds to Example 3(a) section D.4, ISO 11929:2010 |
| ISO-Example-3b\_EN.txp | I-131 accumulation on air filter;  corresponds to Example 3(b) section D.4, ISO 11929:2010 |
| ISO-Example-4\_EN.txp | Activity concentration with peak evaluation (Ge-Detector);  corresponds to Example 4, Section D.5.1, ISO 11929:2010 |
| ISO-Example-5\_EN.txp | Peak net counting rate (NaI(Tl)-Detector); corresponds to Example 5 Section D.5.2, ISO 11929:2010 |
|  |  |
| Michel-2000-b\_EN.txp | I-129 determination in soil by AMS (one value changed) |
| Sterlinski-2008-NAA\_EN.txp | Cs determination in tobacco by neutron activation analysis (**en**) (new example) |
| ISO-Neutron-Dose\_EN.txp | Measuring neutron dose in a mixed radiation field; replaces AKS-Neutron-Dose\_EN.txp |
| ISO-Photon-Dose\_EN.txp | Measuring neutron dose in a mixed radiation field; replaces  AKS-Photon-Dose\_EN.txp |
|  |  |
| Moreno-Sr90\_IAEA-135\_V4\_EN.txp | Sr-90 determination by LSC in a IAEA-135 sample |
| Alpha-IAEA-1401-Kanisch\_V2\_EN.txp | Alpha-spectrometry of Pu-238 in fish |
| vTI-Alpha-Americium\_EN.txp | Alpha-spectrometry of Americium-241 in fish; rather complex due to two interferences: a) in-growing Am-241 from a Pu-241 impurity of the Pu‑242 tracer and b) Am-241 as impurity of the Am-243 tracer; **a corresponding Excel solution confirmed the results** |
| Ra226\_U235-at-186keV\_EN.txp | Gamma spectrometric determination of Ra-226 (186 keV line) with subtraction of a U-235 contribution (interference); **a corresponding Excel solution confirmed the results** |
|  |  |
| DWD\_AB-Gesamt-Aeros-Alpha1\_EN.txp | Monitoring artificial-alpha activity concentration in air, version 1 |
| DWD\_AB-Gesamt-Aeros-Alpha3\_EN.txp | Monitoring artificial-alpha activity concentration in air, version 3 |
| DWD\_AB-Gesamt-Aeros-Beta1\_EN.txp | Monitoring artificial-beta activity concentration in air, version 1 |
| DWD\_AB-Gesamt-Aeros-Beta3\_EN.txp | Monitoring artificial-beta activity concentration in air, version 3 |
| Gamma-Dist\_EN.txp | Application of the (N+1) rule for the case of very low count numbers of gross and background counting rates |
| Lira-GammaDist\_EN.txp | Application of the (N+1) rule for the case of very low count numbers of gross and background counting rates; **example from Lira & Grientschnig, 2010** |
| Fe-55-with-LSC-and-standard-addition\_EN.TXP | LSC measurement of Fe-55 using the method of standard addition  The gross count rate symbol is occurring twice in the calculation equation, in the nominator and in the denominator of that expression. See the description given in the project file.  (**Independently confirmed by D. Schrammel (KIT) and Prof. Michel** ) |
| NLWKN\_Fe-55\_with\_KALFIT\_EN.txp | Fe-55 beta measurement evaluation including a full calibration curve for the counting efficiency |
| Example\_8\_with\_KALFIT\_EN.txp | Example 8 of the new (German) Supplement 1 to DIN ISO 11929 (2014). |
| Mean-theta\_EN.txp | Sr-90-Measurement with several individual measurements of a reference sample and input of its data set to the project, which allows deriving a parameter theta, by which mean und uncertainty can be calculated. |
| TemperaturCurve\_KALFIT\_V2\_EN.txp | Interpolation of a linear temperature calibration curve, **example from JCGM 100:2008**: b(t) = y1 +y2\*(t - t0) |
| BSH\_total-Gamma\_var1\_DE.txp | total Gamma measurement in seawater, version 1 |
| BSH\_total-Gamma\_var2\_DE.txp | total Gamma measurement in seawater, version 2 |
| Ac228\_binomial\_V2\_EN.txp | Measuring a short-lived radionuclide with long counting: binomial distributed sample contribution to the gross counts |
| Ra226\_U235-at-186keV\_EN\_long.txp | Same as the one show above, but with longer symbol names |
| sumEval\_sum\_V3\_EN.txp | Combining 4 measurements by summation for determining one output quantity |
| sumEval\_mean\_V3\_EN.txp | Combining 4 measurements by averaging for determining one output quantity |
| PresetCounts\_EN.txp | Simple single channel measurement for demonstrating the case of measurement with pre-set counts (number of counts n are fixed; counting duration t is variable) |
| A set of projects  iso11929-4\_Example-6\_EN.txp  through  iso11929-4\_Example-17\_EN.txp | Projects prepared for the examples considered in the **standard ISO 11929-4:2022** |
|  |  |
| ***with linear unfolding:*** |  |
| vTI-Y90-16330\_Blw\_V2\_EN.txp | Y-90 decay curve, including blank contribution (fish sample) |
| vTI-Y90-16671\_Blw\_V2\_EN.txp | Y-90 decay curve, including blank contribution (fish sample) |
| vTI-Y90-16748\_Blw\_V2\_EN.txp | Y-90 decay curve, including blank contribution (fish sample) |
| Several-peaks-nuclide-activity-V3\_EN.txp | Activity, determined by weighted mean from several γ-peaks of a radionuclide |
| La140\_REMSPEC-4Lines-V3\_DE.txp | Activity, determined by weighted mean from 4 γ-lines of the radionuclide La-140 |
| Ratel\_Annex1\_Beispiel\_EN.txp | Measurement of the decay curve of Fluor-18 (half-live of 1.829 h); **example from Ratel et al., Metrologia, 2015** |
|  |  |
|  |  |
| **Example with more than 1 output quantity:** | |
| ***without linear unfolding:*** |  |
| Janszen-Sr-89-Sr-90\_V4\_EN.txp | Sr-89/Sr-90 determination in soil/sediment 🡪 **IAEA-1401** |
| J-ALUFT-Sr-89-Sr-90\_V2\_EN.txp | Sr-89/Sr-90 determination in exhaust air |
| Galpha\_beta\_Rusconi\_2006\_V2\_EN.txp | Total alpha- and total-beta determination in water by LSC measurements in two windows, with alpha/beta discrimination |
| dwd\_sr89\_sr90\_TDCR\_procedureV2\_EN.txp | Determination of Beta emitters Sr-89 und Sr-90 by a TDCR-procedure, as realised with a HIDEX LSC Counter |
| ***with linear unfolding:*** |  |
| Sr89-Sr90\_Schrammel\_EN.txp | Sr-89/Sr-90 determination by LSC, with 1 energy window; simple |
| DWD-LSC-3Kanal-V2\_EN.txp | Sr-89/Sr-90 determination by LSC, with 3 energy windows; complex |
| DWD-LSC-3Kanal-V2-fd\_EN.txp | Sr-89/Sr-90 determination by LSC, with 3 energy windows; complex; with using the (decay) function fd() |
| J-ALUFT\_Sr-89\_Sr-90\_Linf\_EN.txp | Sr-89/Sr-90 determination in exhaust air (compare with J-ALUFT-Sr-89-Sr-90\_V2\_EN.txp) |
| LUBW\_Sr-89\_Sr-90\_with-Sr-85-fixed\_V2\_EN.txp | Sr-89/Sr-90 determination with Sr-85 tracer added, where the tracer count rate contribution is NOT subject to fitting |
| Sr89-Sr90\_IAEA\_AQ-27\_2013\_V2\_EN.txp | Sr-89/Sr-90 determination by LSC; energy window and counting efficiencies vary between 1st and 2nd measurement |
| Sr89\_Sr90\_LSC-without-Sr85\_EN.txp | Sr-89/Sr-90 determination by LSC, with 3 energy windows; without Sr-85-tracer; with covariances between window efficiencies |
| Tritium\_4Bubbler\_used\_1-3\_DE.txp | Measurement of HT and HTO in air with applying a 4-fold-Bubbler (according to J.-M. Duda, JER 189 (2018) 235-249), application of linear unfolding; bubblers 1,2 and 3 were evaluated |
| Tritium\_4Bubbler\_used\_2-3\_DE.txp | Measurement of HT and HTO in air with applying a 4-fold-Bubbler (according to J.-M. Duda, JER 189 (2018) 235-249), application of linear unfolding; bubblers 2 and 3 were evaluated |
| Pb210\_Bi210\_Po210\_series\_backwards\_EN.txp | Examples of using the UR function SDECAY for a 3-member radioactive decay chain |
|  |  |
|  |  |
| **Other examples from the literature, evaluated, without detection limits:** | |
| Neutron-Dose-Cox-2006\_V2\_EN.txp | Determination of neutron dose equivalent |
| Calibration-of-weight-Cox-2001\_V2\_EN.txp | calibration of a weight |
| Kessel-1-2006\_EN.txp | calibration of a mass of nominally 10 kg |
| Kessel-2a-2006\_EN.txp | Pb mol mass determination, with several correlations |
| Kessel-2b-2006\_EN.txp | alternative Pb mol mass determination, with several correlations |
| Wuebbeler-Ex1\_EN.txp | MC example for non-gaussian distribution |
| Wuebbeler-Ex2\_EN.txp | MC example for non-gaussian distribution |
| PearsonYork\_with\_KALFIT\_EN.txp | Application of weighted total LS (WTLS) to the data Pearson & York data set |
|  |  |
| **Example for a “negative” linear Model:** | |
| Rn-222-Emanation\_EN.txp | Detection limit calculation for a Rn-222 emanation coefficient |
|  |  |
|  |  |

### 3.3.2 Revision of physical units in the examples

In UncertRadio an option was introduced for deriving the physical units of dependent quantities by calculations; see chapters 2.27, 2.26 und 7.21. In the context of applying this option, the units used in the example projects of section 3.3.1 were tested for correctness. Several modifications were found to be necessary. They mean, that not all units introduced earlier by the author of this program were correct or functional.

In some cases, notes about the modifications were documented within the project file, TAB “Procedure“. In most of the cases, it was necessary to change the existing unit “1” (or “ “) for detection probability variables (often called eps…) to “1/Bq/s“. This helps the output variable to receive the unit part “Bq“ instead of “1/s“. These latter changes were only seldom documented within the project files. In the example J-ALUFT-Sr-89-Sr-90\_V2\_EN.txp, two parameters, a and b, used for calculating eps2, got the new units “1/Bq/s/mg“ and “1/Bq/s“, respectively.

In another case, Ra226\_U235-at-186keV\_EN.txp, the equation RRa = RS – RU5 resulted in a difference of the units “1/s“ and “Bq“; in this case, in the program the first one was then applied as the unit name for RRa.

In the case of Ac228\_binomial\_V2\_EN.txp, the detection probability epsD is used two times, as part of the parameter p, which should be dimensionless as a parameter of the binomial distribution, and as a part of the calibration factor. This dilemma was solved such that the epsD which is used within the expression for p, receives the unit “1“ (or “ “), while a second variable epsD\_U was introduced, which as part of the calibration factor got the unit “1/Bq/h“, but the measurement value was set equal to one (without uncertainty).

In such examples explicitly containing scaling factors of 60 for the unit “min“ or 1/1000 for the unit “g“, two special switching variables (or Trigger variables) were attributed to these factors; see section [2.2.6](#URH_Switching_variables).

For a group of projects their version number (\_Vx\_) in the file name was increased:

Ac228\_binomial\_V2\_EN.txp

DWD\_sr89\_sr90\_TDCR\_procedure\_V2\_EN.txp

Galpha\_beta\_Rusconi\_2006\_V2\_EN.txp

J-ALUFT-Sr89-Sr-90\_V2\_EN.txp

Janszen-Sr-89-Sr-90\_V3\_EN.txp

Moreno-Sr90\_IAEA-135\_V2\_EN.txp

sumEval\_sum\_V2\_EN.txp

sumEval\_mean\_V2\_EN.txp

vTI-Y90-16330\_Blw\_V2\_EN.txp

vTI-Y90-16671\_Blw\_V2\_EN.txp

vTI-Y90-16748\_Blw\_V2\_EN.txp

A consequence of calculating units of dependent variables (withing the menu item “test physical units“) is the transformation to basic units. In some cases, the associated scaling factors change the output quantity value powers of 10:

Galpha\_beta\_Rusconi\_2006\_V2\_EN.txp: factor 1000 (1/g 🡪 1/kg) (permanently changed)

Sterlinski-2008-NAA\_EN.txp: factor 1.0E-9 (due to a unit “ng/g“)

sumEval\_summe\_V2\_DE.txp: factor 1.0E+4 (1/cm2 🡪 1/m2)

sumEval\_mitteln\_V2\_DE.txp: factor 1.0E+4 (1/cm2 🡪 1/m2)

The last three changes do not apply if the test of physical units is not used, i.e., if the program is used in the normal mode!

A new version was prepared for the file with reference values of the example projects:

BatListRef\_v06.txt

## 3.4 Options dialog - Presetting

|  |
| --- |
|  |

This dialog allows the definition of the following parameters used in the calculations of Decision threshold and Detection limit:

* **Quantiles of the normal distribution**,  **and , and their associated probabilities for the errors of the first and second kind, and ,** respectively. The program only works with the values of the quantiles and; it is, however, possible to define first the error probabilities from which in turn the quantiles are determined then.

**Presetting** in the case of preparing a new project:

k-alpha = 1.644854, alpha = 0.05  
k-beta = 1.644854, beta = 0.05.

If a quantile value is modified, the associated probability value must be changed; this is indicated by disabling its editing and the button “adjust“ (above) becomes activatable. In the case of modifying a probability value first, the procedure is just vice versa. By clicking the “adjust“ button, the other associated value is adjusted; without this adjustment the dialog cannot be closed.

* the **method of defining Decision thresholds and Detection limits**: only the method of ISO 11929:2019 is available.
* For the calculation of the limits of the (two-sided) confidence interval (better: coverage interval), the value of the associated probability is needed, which can be given in the corresponding field. The default value is .
* The value of the **coverage factor *k*** is required for calculating the expanded uncertainty, which can be defined in the field **coverage factor output**. Its pre-set value is 1 (internal variable coverf).
* If in a project file the uncertainties of the listed independent input quantities are given as expanded uncertainties, its associated coverage factor needs to be removed while reading them in. This coverage factor can be given in the field **coverage factor input**. Its pre-set value is 1 (internal variable coverin).

Usually, coverin=1 is applied when working with a project within UncertRadio by dialog. The value coverin=2 probably will be restricted to the case when UncertRadio used in an automated way from within another program. An Excel application prepares a project file filled with data within Excel and calls UncertRadio for evaluating it. If the input quantity uncertainties had been entered in Excel with coverin=2,and the parameter coverin=2 is given in the project file, UncertRadio converts these uncertainties to k=1 by dividing them by the coverin value. The internal calculations are all done with a coverage factor of 1. Just before the output to an external CSV file, which then will be imported by Excel, UncertRadio multiplies the uncertainty values with the value of the parameter coverf, which also has to be defined in the project file.

When using UncertRadio in the automated way, care has to be taken about how to correctly insert the uncertainty values into the project file: refer to [chapter 3.6.3](#URH_Hints_input_uncertainties_DE).

* The dialog contains an **input field for a variable GamDistAdd**. This parameter represents the x in the (N+x) rule for counts or counting rates, between 0 and 1. The values 0, ½ und 1 for GamDistAdd correspond to common priors within a Bayesian view, which are proportional to , with *c*=GamDistAdd. For variables for which the (N+x) rule has been selected, a corresponding Gamma distribution is assumed.

|  |  |  |
| --- | --- | --- |
| *c*=GamDistAdd | (1-c) | mean |
| 0 | 1 | N+0 |
| 1/2 | 1/2 | N+1/2 |
| 1 | 0 | N+1 |

* The **language** output within the program and its dialogs and the can be switched between German and English after the program’s start; the list separator character can also be selected. See also: [country specific parameters](#URH_COUNTRYSPECS_EN).

See also: [**Note on a subsequent call of this options dialog**](#URH_AEND_OPTION_EN)

## 3.5 [Advice in case of problems](#URH_FIRSTAID_EN)

**Advice in case of problems**

* If the project runs through the end without any messages, but the results are not plausible, e.g. the detection limit iteration does not converge:

Then check first whether the symbols for the net and gross counting rates are correctly selected.

This means also that always a separate equation must exist which defines the net counting rate.

* The detection limit iteration also starts working improperly if the relative uncertainty approaches a value of 61 % or even fails in case of more than 61 %. The reason for this behavior originates from ISO 11929 itself.
* Also, values in the table of the TAB “Values, uncertainties“, which are equal to zero, lead to strange behavior. This may happen after re-editing the equations, where shifts between the lists of symbol names and values may occur. Thus, the latter should be checked for after one would have done significant re-editing of the equations.
* *Symbols* which *became dispensable* after having edited the equations: it is referred to [section 4.2](#URH_Dispensable_Symbols_EN) for their removal.
* Calls to the special UR specific functions LINFIT or KALFIT should always be defined in separate equations; the closing brackets of these calls must not be followed by additional operators or symbols in the same equation.
* A simple way in UR for initiating a complete re-calculation of the output values consists in selecting the current out output quantity again in the menu Edit - Output quantity.
* The toolbar icon view-refresh.pnginitiates a complete re-calculation of characteristic values after having made changes to the model or the input data. If the program halts at the TAB “Values, Uncertainties”, press the view-refresh.png icon once more.
* Editing cells in tables:

one click into the cell row marks the row, the second click opens the cell for editing. An entry into a cell must be confirmed with the ENTER key; the TAB key is not sufficient.

* Paste the Windows Clipboard into a table cell:

Note that paste with the short-cut CTRL-V only works for the first time, but no longer thereafter.

Thereafter, this works only correctly, after having marked the cell, by using the paste option from the associated context menu;

* To optimize the column widths is possible by double-clicking the small vertical line between two column heads; the mouse pointer changes its shape shortly before double-clicking.
* For importing column blocks from Excel or from Notepad++ into columns of a UR2 table: see the end of the Help chapter 7.7.
* Using the menu item **Edit – Load missing values from project variant**, values and uncertainties still missing in the actual project can be imported from another file variant of the project.

## 3.6 Structure of the project file

Knowing the structure of a project file is useful for a semi-automated usage of UncertRadio. For a regularly repeated Sr-89/Sr-90 analysis, e.g., with linear unfolding and a fixed scheme, one will probably be able to establish a basic version of that UncertRadio project file where some parts in it will (nearly) not change. Another part will change, however, from measurement to measurement. The latter data may e.g. be written directly into that file using a small Visual Basic program (MS Excel).

In addition to the existing version of a project file as .txp text file the structure of which is described below, a CSV file version can be used; the latter will be discussed at the end of this help topic.

### 3.6.1 Project file as text file in \*.TXP format

This may done by first copying the basic version of the project file to a working project file, also with extension .txp, e.g. to “Test2.txp”. Only when passing special keywords (those with preceding @, see below), those data from the actual measurement are written into that working file. After closing the working file one can call UR e.g. directly from a Visual Basic program such that UR is initially loading just this edited project file, by some command like “start uncertradio.exe Test2.txp”. Such as call can also be done like:

D:\UR > Uncertradio.exe Test2.txp

The table following below shows the simple structure of a project file (extension TXP).

Note: numbers always with . as decimal point (no comma!)!

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Deutsche Beschreibung: | | | | | | | | | English description: | | | |
| **Project:** Name der CSV-Projektdatei / name of CSV project file | | | | | | | | | | | | |
| **@Titeltext:** | | | | | | | | | | | | |
| Text zur Beschreibung des Verfahrens | | | | | | | | | text describing the procedure | | | |
| **@Formeltext:** | | | | | | | | | | | | |
| es folgen die Gleichungen zur Berechnung der Aktivität | | | | | | | | | is followed by the equations for calculating the activity | | | |
| **@FormeltextFit:** | | | | | | | | | | | | |
| es folgen die Gleichungen der energieabhängigen Funktionen im Verfahren *Gamspk1* oder der Abklingfunktionen im Verfahren *Linfit1*. | | | | | | | | | is followed by equations describing the energy dependent curves in the method *Gamspk1* or the decay functions in the method *Linfit1*. | | | |
| **@Symbole-GRID:** | | | | | | | | | | | | |
| folgende Variablen werden Werte zugeordnet: | | | | | | | | | values are attributed to the following variables: | | | |
| nchs= 1  nEGr= 1  ngrs= 8  nab= 3  nmu= 5 | Anzahl der Messkanäle  Anzahl der Ergebnisgrößen  Anzahl der Variablen (Symbole)  Anzahl der abhängigen Symbole (a)  Anzahl der unabhängigen Symbole (u) | | | | | | | |  | number of counting channels  number of output quantities  number of variables (symbols)  number of dependent symbols (a)  number of independent symbols (u) | | |
| Hiernach folgen zeilenweise die Informationen, die zum „Symbole-Grid“ im TAB „Gleichungen“ gehören, für jedes Symbol (in den Zeilen dient das Zeichen # als Trennzeichen):   * der Name, * der Typ (a oder u), * die Einheit und * die „Bedeutung“; | | | | | | | | | For each symbol from the „symbol grid” of the TAB “Equations” following information is given (with # as separating character):   * name, * type (a or u), * unit and * the symbol‘s „meaning“; | | | |
| **@Menu1 und Menu2:** | | | | | | | | | | | | |
| Nummern der Symbole der: | | | | | | | | | numbers of the symbols of: | | | |
| knetto= 2 | | Nettozählrate | | | | | | | net counting rate | | | |
| kbrutto= 0 | | Bruttozählrate (0: nicht verwendet bei Verfahren mit Entfaltung) | | | | | | | gross counting rate (0: not used with procedures using linear unfolding) | | | |
| **@Unc-Grid:** | | | | | | | | |  | | | |
| Es folgt für jede Variable eine Zeile, in der mit einer csv-Struktur (# als Trennzeichen) folgende Elemente (Zahlenwerte: double precision; -999.d0 bedeutet „nicht definiert“) stehen: | | | | | | | | | for each variable the following elements are given in one row (with a csv structure; # as separating character; numbers: double precision, -999.d0 means “not defined”: | | | |
| * Symbol; * Wert der Größe; * Index des Verteilungstyps:   1 Normalverteilung  2 Rechteckverteilung  3 Dreiecksverteilung  4 (N+x)-Regel für Impulsanzahlen  5 Lognormalverteilung  6 Gammaverteilung  7 Binomia+Poisson-Verteilung  8..2-ParameterBetaverteilung  9.. t-Verteilung   * Text-Formel für die Standardunsicherheit; * Zahlenwert der Standardunsicherheit; * Halbreite der Verteilung (Dreieck oder Rechteck); * 1: absolute bzw. 2: relative Unsicherheitsangabe bei der Eingabe; * berechnete absolute Standardunsicherheit. | | | | | | | | | * symbol; * value of the quantity; * Index of the distribution type   1 normal distribution  2 rectangular distribution  3 triangular distribution  4 (N+x) rule for counting numbers  5 lognormal distribution  6 gamma distribution  7 binomial+poisson distribution  8 2-parameter beta distribution  9..t-distribution   * text formula for standard uncertainty; * value of the standard uncertainty; * Half-width of a distribution (triangular or rectangular); * interpretation of the uncertainty:   1: absolute, 2: relative;   * calculated absolute standard uncertainty. | | | |
| **@Covar-Grid:** | | | | | | | | | | | | |
| Falls Kovarianzen definiert sind, folgen diese hiernach folgt für jedes Variablenpaar:   * die zwei Symbol-Nummern des Variablenpaars; * 1 für Kovarianz oder 2 für Korrelation; * ggf. Textformel für die Kovarianz; * Wert der Kovarianz (1) bzw. Korrelation (2) | | | | | | | | | If covariances are defined, they are characterized for each pair of variables:   * the two symbol numbers of the pair; * 1 for covariance, or 2 for correlation; * if necessary, covariance formula as text; * value of covariance (1) or correlation (2) | | | |
| **@Abkling-Grid:** | | | | | | | | | | | | |
| ModPar= 1 1 1 1 1 0 0  8 Werte (8 values):  #1 -3: ifit(.)  #4:  #5:  #6:  #7:  #8:  01.01.2005 12:12:07  1 | | | 7 checkbox/radiobutton-Werte im Grid;  welche Parameter fitten? (1: fitten; 2: festhalten; 3: weglassen)  gewichteter Fit?  Benutzung Kovarianzen  Wahl der Fitmethode:  0: WLS, 1: PLSQ,  2: PMLE, 3: WTLS  WTLS: nein/ja (jetzt überflüssig)  Nachweiswahrscheinlichkeiten variabel zwischen Messungen:  0: nein; 1: ja  Datum/Uhrzeit Sr/Y-Separat.;  Zeiteinheit (1: s; 2: min). | | | | | | 7 checkbox/radiobutton values within the grid defining the model of the decay curve;  which parameters to fit? (1: fit; 2: fix it; 3: omit it)  use weighted fit?  use covariances?  use of fitting method:  0: WLS, 1: PLSQ,  2: PMLE, 3: WTLS  WTLS: no/yes (now redundant)  counting efficiencies vary with measurements:  0: no; 1: yes  Datetime of e.g. Sr/Y separation:  time base (1: s; 2: min). | | | |
| Hiernach folgen, für jede Messung der Abklingkurve eine Zeile, die folgende Daten enthält:   * Startdatum der Messung – oder Zeitdifferenz zum Abtrenn-Datum; * Messdauer der Bruttomessung; * Bruttoimpulse; * Bruttozählrate; * Unsicherheit der Bruttozählrate; * Messdauer der Nulleffektmessung; * Impulse der Nulleffektmessung; * Nulleffektzählrate; * Unsicherheit der Nulleffektzählrate; * Nettozählrate; * Unsicherheit der Nettozählrate; | | | | | | | | | Hereafter, for each measurement of the decay curve one row is given, containing following data:   * start date of the measurement – or time difference to datetime of the chemical separation; * gross counting duration; * gross counts; * gross counting rate; * uncertainty of gross counting rate; * counting duration of background meas.; * counts of background measurement; * background counting rate; * uncertainty of background counting rate; * net counting rate; * uncertainty of net counting rate; | | | |
| **@Gamspk1-Grid:** | | | | | | | | | | | | |
| CurveUse= 0 0 0 0 | | | | für Checkbox-Werte im Dialog „Definition energieabhängiger Kurven“ (nur UR1-Projekte); wird nicht mehr verwendet; | | | | | for checkbox values within the dialog „Definition of energy dependent curves“(only UR1 projects); no longer used; | | | |
| UnitRadio= ***1*** 1 ***1*** 2 ***1*** ***1*** ***1***  Reduced to, since V. 2.4.19:  UnitRadio= ***1*** ***1*** ***1*** ***1*** ***1***  *(see section 7.16)* | | | | Werte der Radiobuttons (1 oder 2) im Dialog zur Eingabe der Gammazählraten | | | | | Radio button values (1 or 2) within the dialog for input of counting rates of gamma lines | | | |
| UnitRadio= 1 1 1 2 1 1 1 | | | | Werte der Radiobuttons im Dialog zur Eingabe der Gammazählraten | | | | | radiobutton values within the dialog for input of counting rates of gamma lines | | | |
| MeanTyp= 1 | | | | Mittelwerttyp | | | | | type of mean used for averaging activities | | | |
| FBT= 1.1200 | | | | (1+b/(2L))-äquivalenter Faktor | | | | | factor equivalent to (1+b/(2L)) | | | |
| EcorrUse=1 | | | | sollen Effizienz-Korrelationen verwendet werden? | | | | | shall peak efficiency correlations be used? | | | |
| Hiernach folgen, für jede Gammalinie eine Zeile:   * 1 für Verwendung der Linie, sonst 0; * Energie der Linie (keV); * Wert der Nettozählrate, Rnet; * Wert der Untergrund-Kontinuum-Zählrate, RT; * Wert der Netto-Nulleffektzählrate, Rbg im separaten Spektrum; * Wert der Effizienz, effi; * Unsicherheit der Effizienz; * Gamma-Emissionsrate, pgamm; * Unsicherheit von pgamm; * Wert der Selbstschwächungskorrektion, f\_att; * Unsicherheit von f\_att; * Wert der Koinzidenzsummationskorrektion,  f\_coin; * Unsicherheit von f\_coin * Ein früher noch vorhandener Parameter WMextSD wird nicht mehr verwendet | | | | | | | | | Hereafter, for each gamma line one row:   * 1 for using the line, 0 otherwise; * energy of the line (keV); * value of net counting rate; * value of the background continuum counting rate, RT; * value of net counting rate from separate background spectrum, Rbg; * value of the peak efficiency, effi; * uncertainty of effi; * Gamma-emission rate, pgamm; * uncertainty of pgamm; * value of self-attenuation correction, f\_att; * uncertainty of f\_att; * value of coincidence summing correction,  f\_coin; * uncertainty of f\_coin * a parameter WMextSD used earlier is no longer considered | | | |
| **@Kalfit-Grid:** | | | | | | | | | | | | |
| KalPars= 5 2  #1: 5  #2: 2  Ctitel | | | | | | 2 Integer-Feld-Werte im Dialog:  Anzahl der Kalibrierpunkte  Grad des Fit-Polynoms (0-3)  (0: Mittelwert der Y-Werte)  Titel der Kalibrierkurve | | | | | | 2 integer field values in the dialog:  number of calibration points  Degree of fit polynomial (0-3)  (0: Mean of the y-values)  Title of the calibration curve |
| Hiernach folgen zeilenweise 4 Werte der Kalibrierkurve:  X-Wert,  Std Unsicherheit des X-Werts (oder leer oder 1),  Y-Wert,  Std Unsicherheit des Y-Werts (oder leer oder 1)  (1 bedeutet identische Wichtung) | | | | | | | | Hereafter, 4 values of the calibration curve follow, line by line:  X value,  Std uncertainty of X value (or empty or 1),  Y value,  Std uncertainty of Y value (or empty or 1)  (1 means identical weighting) | | | | |
| **@DChain:** | | | | | | |  | | | | | |
| CHName=Pb-210-3N Name der Zerfallsreihe | | | | | | | Name of the decay chain | | | | | |
| Pars= 3 1 1 1 0 0 3 | | | | | | |  | | | | | |
| 3: index of chain selected from the file | | | | | | | 3: index of chain selected from the file | | | | | |
| 1: chemische Separation (0: nein; 1: ja) | | | | | | | 1: chemical separation (0: no; 1: yes) | | | | | |
| 1: verwende lambda anstatt T12 (0: nein; 1: ja) | | | | | | | 1: use lambda instead of half-live (0: no; 1: yes) | | | | | |
| 1: Anzahl Energiefenster (Kanäle) | | | | | | | 1: number of energy windows (channels) | | | | | |
| 0: gemeinsame Messung (0: nein; 1: ja) | | | | | | | 0: common measurement (0: no; 1: yes) | | | | | |
| 0: Nachwachsen der Tochter nach chem. Seperation (0: nein; 1: ja) | | | | | | | 0: ingrowth of daughter after chem. Separation (0: no; 1: yes) | | | | | |
| 3: Anzahl der Glieder der Zerfallsreihe | | | | | | | 3: number of members of the decay chain | | | | | |
| Darstellung der Zerfallsreihe der Datei: | | | | | | | Representation of the chain in a file: | | | | | |
| 1 #Pb-210 #lamPb210 #epsPb210 # # #etaPb # | | | | | | |  | | | | | |
| 2 #Bi-210 #lamBi210 #epsBi210 # # #etaBi # | | | | | | |  | | | | | |
| 3 #Po-210 #lamPo210 #epsPo210 # # #etaPo # | | | | | | |  | | | | | |
|  | | | | | | |  | | | | | |
| **@Sonstige:** | | | | | | | | | | | | |
| kalpha=3.000000  kbeta=1.644854  coverf=1.000  coverin=1.0  1-gamma=0.9500  GamDistAdd=0.0  ModelType=PosLin  BinPoi=8 10 12 9  **Note:** NWGTyp is no  longer used | | | Quantil zum Fehler 1. Art α;  Quantil zum Fehler 2. Art β;  Erweiterungsfaktor für die Ergebnisunsicherheit;  Erweiterungsfaktor für die Eingabeunsicherheit unabh. Größen;  Wahrscheinlichkeit zum Vertrauensintervall  Parameter für Gammaverteilung;  ModellTyp  (one of: PosLin/GUMonly/NegLin);  Binom/Poisson-Fall: Symbol-Nummern von p, R0, tm, lambda | | | | | | Quantile associated with type I error, α;  Quantile associated with type II error, β;  coverage factor for uncertainty of the output quantity;  coverage factor for uncertainties of the independent input quantities;  probability associated with the confidence interval  Parameter for Gamma distribution;  Type of model (one of PosLin/GUMonly/NegLin);  Binom/Poisson case: symbol numbers of p, R0, tm, lambda | | | |
| **@Means:** | | | | | | | | | | | | |
| meantyp= 2 2  refmean= 2 (> 0)  gefolgt von / followed by:  n Zeilen der Art: / n lines like:  „name\_data 1.5 2.0 2.2…“ | | | | | eine Folge von n Mittelwert-Typen (integer, Werte 1, 2 oder 3) der n Mittelwert-Variablen  Bedeutung der meantyp-Nummer:  („unbekannte zufällige Einflüsse“)  1: siehe Gl. (1) in 6.9.1  2: siehe Gl. (3) in 6.9.1  3: siehe Gl. (5) in 6.9.1  für Referenz-Zweck selektierter Datensatz  „bekannte Einflüsse“: siehe Gl. (6) u. (7)  in 6.9.1  Ein Variablen-Bezeichner, gefolgt von den Einzelwerten der Variablen mit dem Namen „name“ aus der Symbolliste | | | | | | a series of n types of means (integer, values 1, 2 or 3) of the n mean variables  Meaning of meantyp number:  (“unknown random influences”)  1: see Eq. (1) in 6.9.1  2: see Eq. (3) in 6.9.1  3: see Eq. (5) in 6.9.1  data record selected for reference purpose of  “known influences”: see Eqs. (6) and (7) in 6.9.1  a variable-identifier, followed by the single values of the variable with the name “name” of the symbol list | |

### 3.6.2 Project file in the Excel-compatible \*.CSV format

The structure of the CSV format is similarly to that of the \*.TXP format. It is also included in the preceding table describing the structure of the txp format. The following figure shows an example. **@Sonstige** from the txp format is found under **Optionen**.

**NOTE**: If one loads a TXP project file into UR and saves it in CSV format, one should not edit this CSV file with Excel and saves it again from within Excel. The reason is that Excel saves this CSV file by modifying the real numbers in it by reducing their numbers of decimals, which may lead then to slightly different results when loading this file again into UR. Such a manual editing of the CSV file should better be done by using a simple text editor such as Notepad.

As the automated usage of UR can be done via an Excel application, it could be confirmed that this problem does not occur when using the export to CSV from within Excel via an VBA code.





### 3.6.3 Notes about the input of input quantity uncertainties

It is to be noted that uncertainties of input quantities can be entered in different fields. These are given in the following table, which corresponds to the TAB “Values, uncertainties”.

Using the \*.TXP format of the project file, these fields are found under the section @Unc-Grid: described under the 6th to the 9th bullet; in the CSV format these are the columns 5 – 8 (E – H) under the section “Values, uncertainties“.

An uncertainty values must not be entered in column 10; dependent on the distribution types, it has to be entered in one of those columns given as the columns 6 to 9 in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| 6: | **StdDev formula** | text field | formula of the standard deviation of the quantity; no formula if (N+x) rule has been selected;  (the internal coverage factor is always 1); always “.” has to used for the decimal point |
| 7: | **StdDev value** | number field | value of the uncertainty for normal distribution;  if the (N+x) rule has been selected, **nothing** shall be entered in these cells!  (the internal coverage factor is always 1) |
| 8: | **Half width** | number field | Half width of rectangular/triangular distribution  (the internal coverage factor is always 1) |
| 9: | **abs./rel.** | select. field | select whether the uncertainty from col. Sp. 6, 7 or 8  are to be taken *absolute* or as *relative* value. |
| 10: | **abs. std.Unc.** | number field | (combined) absolute standard uncertainty calculated by the program form the values of the columns 6, 7, 8 and 9;  Note: a value entered by the user will always be over-written by the program! |

If values are entered in columns 6 to 8 with a coverage factor unequal to 1, the parameter coverin within the same project file must be set equal to the coverage factor just applied. UncertRadio then converts all input uncertainties to the internal coverage factor 1; after the calculations, UncertRadio multiplies the output uncertainties with the value of the parameter coverf defined in the project file.

## 3.7 Font and colors

Usually, so-called CSS file („cascaded style sheets“, often rather complex) are applied for adjusting character fonts and colors. For UR this is reduced to a shortened file **Settings.ini** (belonging to GTK+3). The latter is configured for Windows and contains only two entries.

File **Settings.ini**:

[Settings]

gtk-theme-name = win64

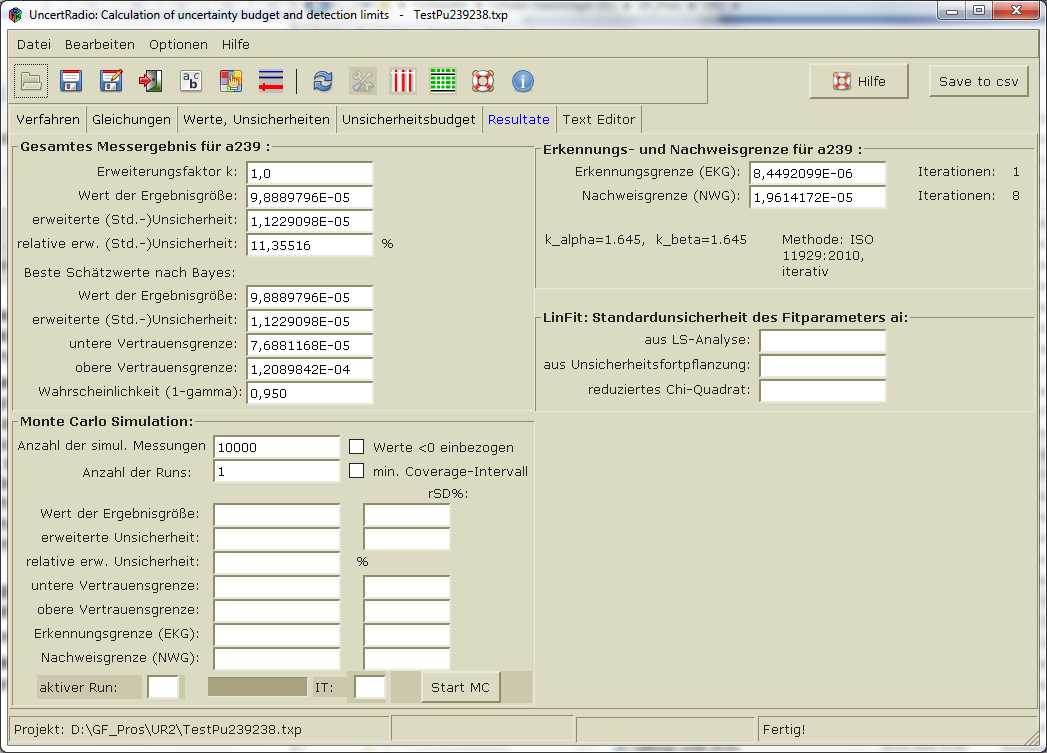
gtk-font-name = Sans 11

Note that the strings to the left of the = characters must not be modified. On starting the UR program the Settings\*.ini file is loaded. This file shall be part of the UncertRadio.exe path.

For modifying values for *gtk-font-name* the **Fontname Icon** of the toolbar can be used.

With the **fontname icon ** the font type and/or the font size can be modified easily. It has to be considered that by enlarging the font size the program’s window also increases.

With the dialog button “**Apply**“ a selected fontname is applied to the UR window. If the font is considered acceptable, it can be saved in the file Settings.ini with the dialog button “**Save**“; this however requires that this fontname had been applied once to the program window. The new fontname will not be saved if this dialog is instead closed by the button “**Quit**”.

*Note: at present, the color button cannot yet be applied.* The **Color icon**  allows choosing the colors of the *permanent dialog background* or the *selected-background* (the latter indicating that a dialog object got the focus). The color selection itself is initiated by clicking the larger rectangular dialog area containing the color button. In a further dialog a color may be selected from a color palette. By clicking the + sign another dialog opens supporting the mouse-driven color selection. At the top of this dialog a text field is shown, e.g. with a string like “#E7E3BA“, which displays the hexadecimal representation of the selected RGB color; in a field to the left of this entry, the currently selected color is displayed. Such a color code can also be entered directly into the text field, which enables a fine-tune “search“ for the color. The hexadecimal code is the one, which is saved with “**Save**” in one of the Settings.ini files. The application of the dialog buttons “**Save**“, “**Apply**“ und “**Quit**“ is the same as for the fontname dialog.

## 3.8 Graphics window

Graphical representations are now displayed in a window containing three Tabs for applications:

* Monte Carlo-Simulation (TAB MC),
* an application not yet „given free“ (TAB MCMC) and
* a graphic related to linear unfolding displaying measured values, standard uncertainties and the fitting curve (TAB LFIT).

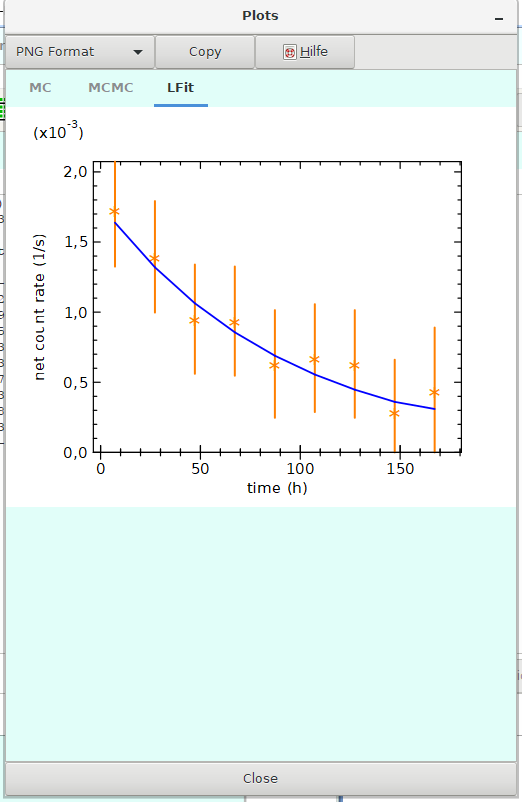
The associated graphical representations are saved as PNG files:

* MCplotfile.png
* -
* CurvePlot.png

The button “Copy“ allows to save the graphic as a file in a format which has been selected with the combobox, where the user is asked for the filename.

This window can be closed with the button “Close“ at the end of the window.

The graphic associated with the TAB LFit can be invoked by the icon  in the toolbar.



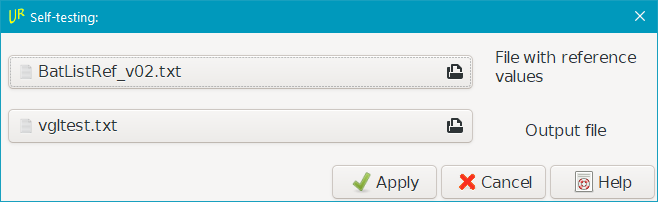
## 3.9 Program testing

After having installed a new program version, there may be a need for checking whether the analytical procedures applied by UncertRadio can be expected to produce evaluation results agreeing with documented results (reference values). This is possible as described now. An already longer existing (internal) test routine had been used to run the many projects, as given in section 3.3, in a batch processing mode for comparing the results with those documented earlier. This routine, previously with no access by the user, has been extended such that it can be invoked now under the **Menu Options – QC batch test**. It tests the analytical procedures for about 2x70 projects (DE+EN), which takes a time of about 40 seconds.

This test requires the reference data file

BatListRef-v06.txt

which is now part of the UncertRadio setup procedure. The dialog invoked via **Menu Options – QC batch test** allows to select the reference data file and an output file showing the comparison of actually calculated and of reference values, project by project.



After the test has finished, information is given about the number of projects for which a disagreement was found. The details for those projects are given in the output file. This file normally is very short, as only those projects are given there for which deviations were found.

# Handling the individual TABs

## 4.1 TAB “Procedure”

The measurement problem to be considered can be characterized here by a short description of the method’s principle and special characteristics may be pointed at. Cited literature may also be included. If the project is under development regarding the equations, this TAB may be skipped.

When the text input is finished one may change by mouse click to the TAB “Equations”.

## 4.2 TAB “Equations”

1) One must begin with the first enabled ASCII text field (the uppermost, the only one enabled when beginning with a new project), in which equations are to be inserted line-by-line: [**Text field for equations**](#URH_GLEICHUNGEN_EN)

2) If all equations have been given formula symbols are extracted from them by clicking the **button** **"Load symbols(1) from equations"**, followed by a syntax check using the internal function parser and the symbols are loaded into a table which then may be edited and values in the columns *Unit* and *Meaning* may be added.

After having edited the equations, the following cases may occur regarding already existing symbols:

a. New symbols have been introduced. The corresponding symbol table rows are green colored.

b. Already defined symbols are no longer needed (**dispensable symbols**). The corresponding symbol table rows are put to the end of the table being yellow colored.

The yellow-colored table rows must at first be marked/selected with the mouse; then, they are removed using the toolbar icon **„delete rows“ ** . Some symbols, not directly used within the equations, but still needed (e. g., counting durations), must not be deleted.

For further editing: see [**Edit symbol list**](#URH_SYMBOLLISTE_EN)

For such quantities within the symbol list, for which mean and variance are be determined from an associated data set, the value “m“ is set in the column “type“ instead of “a“ (dependent quantity) or “u“ (independent quantity). Furthermore, if a quantity, to be calculated by an equation, shall be used as a parameter without uncertainty, the type value “p“ can be used.

3) After the symbol list has been completed (in the testing phase units and meanings need not yet to be defined) by a mouse click on the **button “Load symbols(2) of the finalized symbol table”** the complete symbol list is loaded for the calculations to be done.

4) Now, the two **selection fields** “**Net count rate**” and “**Gross count rate**” are enabled. They allow selecting the appropriate symbols for the net counting rate and the gross counting rate. For the further course through the program **these selections are only of interest to the calculations of Decision threshold and Detection limit**. After having selected them they have to be loaded by clicking the button “Accept net and gross count rates”.

If more than one output quantity is defined, those symbols for the net and gross counting rate must be chosen belonging to the actually active output quantity.

Note: The identification of the net counting rate (more precisely: of the [**procedure dependent net counting rate**](#URH_VERFAHREN_EN)) is needed to allow within the iteration for the Detection limit a simple numerical inversion of the equations, from the varied output quantity (e.g. activity in Bq/kg wet weight) back to the net counting rate (in ). The gross count rate symbol must be identified because the user has to supply the uncertainty function for it; for the determination of Decision threshold and Detection limit the value of the gross count rate is varied internally where this uncertainty function is then used to calculate its associated modified uncertainty value.

**Exceptions:**

a) in the analysis of a decay curve with linear Least squares fitting (calling ***Linfit***) **the indication of a gross counting rates is not required**;

b) similarly, in the case of estimating a mean of activities from individual gamma lines of a radionuclide (calling ***Gamspk1***); the **selection of the gross counting rate is inapplicable**; the **net counting rate**, however, **has to be selected** **because it acts as placeholder for the source activity (Bq);**

c) when summing up or averaging several aliquot measurements (calling **SumEval**)

With these exception cases the symbol to be selected as net counting rate (or the source activity in the case of *Gamspk1*) must be that one which is defined in the equations by the call to *Linfit* or to *Gamspk1*.

5) After step 4 the TAB has been successfully treated and the next TAB to be treated, “Values, uncertainties” is enabled and selected then by a mouse click.

## 4.3 TAB “Values, Uncertainties”

A major part of this TAB consists of a 10-column table for the input of measurement values and uncertainties of the independent measured quantities (input quantities). The columns are:

|  |  |  |  |
| --- | --- | --- | --- |
| 1: | **Symbol** | text field | formula symbol |
| 2: | **Type** | text field | Parameter type: **a** for dependent, **u** for independent, **m** for a quantity mean, **p** for a parameter to be calculated, without uncertainty |
| 3: | **Unit** | text field | input of unit |
|  |  |  |  |
| 4: | **Value** | number field | value of measured quantity |
| 5: | **distribut.** | select. field | selection of the quantity’s distribution  (normal / rectangular / triangular / others)  The [**(N+x) rule for very low count numbers**](#URH_NP1REGEL_EN) may be selected here, implying Gamma distributions of the associated counting rate variables |
| 6: | **StdDev formula** | text field | formula of the standard deviation of the quantity; no formula if (N+x) rule has been selected;  (the internal coverage factor is always 1); always “.” has to used for the decimal point |
| 7: | **StdDev value** | number field | value of the uncertainty for normal distribution;  if the (N+x) rule has been selected, **nothing** shall be entered in these cells!  (the internal coverage factor is always 1) |
| 8: | **Half width** | number field | Half width of rectangular/triangular distribution  (the internal coverage factor is always 1) |
| 9: | **abs./rel.** | select. field | select whether the uncertainty from col. Sp. 6, 7 or 8  are to be taken *absolute* or as *relative* value. |
| 10: | **abs. std.Unc.** | number field | (combined) absolute standard uncertainty calculated by the program form the values of the columns 6, 7, 8 and 9;  Note: a value entered by the user will always be over-written by the program! |

The columns 1 to 3 have been taken from the preceding symbol list and are disabled here.

For the user’s support **those cells in the table are colored with red background which are not filled in by the user**: these are cells belonging to dependent quantities. Only cells with white background should be used for direct input by the user. That means, the rows belonging to dependent quantities are found in the uppermost part of the table, **the first row (row 1) is the one for the output quantity, or the rows 1-3 in the case of three output quantities**.

Special feature of non-normal distributed numbers of counts and count rates: [see](#URH_Treatment_Counts).

For a data set-based quantity, mean and uncertainty are calculated internally from the associated data set and transferred to the “values, uncertainties” table with the button “Calculation of uncertainties”. The corresponding rows in that table therefore have also a red colored background.

The “white cells” in columns 4 to 9 must be filled in by the user. **Numerical values must be input into the column “value”** while this is not always necessary in columns 6 to 9 for each of the quantities. One can often abstain from giving uncertainties, for instance for counting times. The input of uncertainties is allowed in only one of the columns 6, 7 or 8. The selectable field “abs./rel.” however must be used, if in the columns 6 to 8 an entry exists.

For counting rates or numbers of counts **the standard deviation can be input as a formula** in column 6 (“StdDev formula”) - without preceding equal sign. For **quantities declared as normal distributed** - also possible for counting rates/numbers of counts - the numerical value of the uncertainty is input in column 7 (“StdDev value”). In the case of **rectangular and triangular distributed quantities** the uncertainty is characterized by the value of the half-width in column 8. The latter value is internally converted to a standard deviation (normal distribution) according to the GUM rules which then is transferred to the cell in column 10 (“abs. std.Unc.”).

**One cell** within the column “StdDev formula” **is** **highlighted by green colour.** Here, the **standard deviation formula of the gross counting rate** must be given. This formula is required later for the numerical-iterative calculation of the Decision threshold and the Detection limit. It is considered as **“uncertainty function” of the gross counting rate (and implicitly of the (procedure dependent) net counting rate)** **which allows estimating its uncertainty for any varied (iterated) value of the gross counting rate.** It has, however, no meaning for the calculation of the combined uncertainty of the output quantity.

**Possible formulae of the standard deviation of the gross counting rate Rg** (incomplete):

sqrt(Rg/tm) counting devices (counting time tm), single measurement

sqrt(Rg/tm/n) counting devices (counting time tm), n-times repeated measurement

sqrt(Rg/2/tau) ratemeter measurements (time constant tau)

If in addition to the gross count rate Rg the **gross count number Ng** is also used, e.g., by defining an equation Rg=Ng/tm, one should have in mind that **the program also modifies Ng if Rg** **is modified**. From this equation, Rg=Ng/tm, UncertRadio can identify the count number Ng belonging to Rg. This ensures, that the program, during evaluating the equations from the bottom upwards (Ng then is an independent quantity and Rg is dependent), does not overwrite the Rg value and its uncertainty primarily modified, by re-calculating Rg and its uncertainty from Ng and its uncertainty, if the latter would not have been modified also.

If the values of the **gross count rate Rg is obtained as a mean of several individual values,** a linear interpolation is necessary between its variance at measurement (u\_Rgm^2) and the variance to be used for calculating the decision threshold (u\_R0^2). The formula for the standard uncertainty may be expressed as:

sqrt( u\_R0^2 + (u\_Rg^2 – u\_R0^2)\*(Rg - R0) / (Rgm – R0) ) (1)

Herein, Rg und Rgm are the modified (varied) and the measured value of the gross count rate, respectively. If Rg takes the upper value, Rgm, the result is sqrt(u\_Rbm^2), while the result is sqrt(u\_R0^2) for the lower value Rg=R0. This requires appending the symbols Rgm, u\_Rgm and u\_R0 to the symbol list (TAB “Equations”) of independent quantities. They do not get an uncertainty and their values are to be given explicitly in the “Values, uncertainties” table.

**Note:**

It is no longer necessary that the user supplies the equations (1) or a similar formula to UncertRadio. The introduction of additional auxiliary quantities also can be omitted. The chapter 6.9 generally informs how to supply datasets for calculating means. Based on that, the chapter 6.12 describes the necessary calculations and manipulations within the program for deriving an extended version of equation (1).

The type of equation (1) for the uncertainty of the gross count rate-related mean depends on how the mean of single values suffers by additional random influences, which can be characterized as **unknown** or **known**. According to these two options, different versions of Eq. (1) are applied. This is also considered in chapter 6.12, where corresponding example projects are also indicated.

It may happen with special measurement problems that the length of the standard deviation formula of the gross counting rate is longer than the visible part of that cell. For better editing, one can then copy this formula by copy and paste (in the mouse context menu) from this cell to the longer text cell (“**extra working cell for formulae editing**”) located above the table and back again.

Below the uncertainty table is located a smaller **table for the input of covariances**. Its columns are as follows:

|  |  |  |
| --- | --- | --- |
| 1: | **Symbol A** | symbol list box for quantity A |
| 2: | **Symbol B** | symbol list box for quantity B |
| 3: | **Type** | list box for choosing input between covariance and correlation coefficient |
| 4: | **Formula** | text field for defining the covariance as formula being a function of already defined symbols |
| 5: | **(or) Value** | number field for input of the value of the covariance / correlation coefficient |

In the columns 1 and 2 the symbols of correlated measured quantities are selected. After having selected “covariance” in column 3 a formula for the covariance may be entered in the column “Formula”. Otherwise, a numerical value of the covariance / correlation coefficient can be entered directly into the column “(or) Value”.

Relation between correlation coefficient ***r*** and covariance ***cov***:

Having completed the input to the uncertainty and to the covariance tables a mouse click on the **button “calculation of uncertainties”** will initiate the following calculations:

 Note: All calculations within this TAB and the following refer to the actually selected output quantity, if more than one output quantities have been defined for the project. Under the menu item “Edit – Select output quantity” another output quantity may be selected;

 **Uncertainty table**: values of dependent quantities (red colored fields) are calculated and inserted into the corresponding cells; Formulae for standard deviations in the column “StdDev formula” are evaluated; standard uncertainties of independent quantities are evaluated into column “abs. std.Unc.“;

 **Covariance table:** evaluation of covariance formulae as numerical values into column “(or) Value”;

 **Uncertainty table:** now all variances/covariances are known for the complete uncertainty propagation; calculation of the standard uncertainties of the dependent quantities (a) below the output quantity and of the output quantity (row 1 in that table); the combined standard deviations of the dependent quantities (red colored) are exclusively calculated from the uncertainties/covariances of the independent quantities (white colored).

**Only after finalisation of these calculations the TAB “Uncertainty budget” is enabled.**

With more complex measurement problems and a more slowly PC it may be that the calculations take few seconds; it is indicated in the **status bar segment at the lower right corner of the UncertRadio window** with the entry “calculating…” that is still working. After termination of the calculations the entry within this field will be “Ready!”.

**Note:** [**Implication of changing parameters within the Options menu**](#URH_AEND_OPTION_EN)

**For working with tables**: see [**Menu Edit - Table**](#URH_TABTRICKS_EN)

## 4.4 TAB “Uncertainty Budget”

Under this being selected by mouse click the uncertainty budget is presented as a table. The output quantity is indicated which this uncertainty budget is referred to.

The **table “Uncertainty budget”** again contains the three columns “Symbols”, “Type” and “Unit” already known.

The columns “Value” and “Std. uncertainty” contain the input values of the measured value and its standard uncertainty of each of the independent measurement quantities as well as of each dependent quantity (auxiliary and output quantity).

In the column **“Sensitivity coefficient”** partial derivatives of the function of the output quantity ***y*** (i.e., the function determined by the equations with which the value of the output quantity is calculated) are given for each independent quantity.

From the **products Uncertainty x Sensitivity coefficient**, the values of which are shown in the next column, the uncertainty budget is deduced in two different ways.

The column **“relat. contribut(%)“** shows for each independent symbol - in case its uncertainty is non-zero - the relative contribution (in %) of its variance to the total variance of the output quantity. This column gives the information which of the (independent) quantities contributes at most to the combined uncertainty of the output quantity. The indication of 100% in this column for the output quantity is only that the control sum of the individual relative contributions. Using the **button “Change budget type”** allows to display absolute uncertainty contributions in this column, given in the unit of the output quantity.

The preceding **definition of the “relat. contribut(%)“** to the uncertainty budget **has got a further plausible meaning by the recent publication by** **Kessel, Kacker and Berglund** (2006) with the title “Coefficient of contribution to the combined standard uncertainty“:

That relative contribution of an input quantity , divided by 100, is in the case of un-correlated input quantities identical with the square of the correlation coefficient between and the output quantity ***y***! This quantity is now called “**coefficient of contribution”** and is represented by the symbol .

The generalised definition of the “coefficient of contribution” is now:

(1)

As we have for non-correlated input quantities:

(2)

from Eq. (1) then follows Eq. (3), which was already known - for non-correlated input quantities - as the positive relative contribution to the variance of the output quantity:

(3)

If correlations between input quantities exist, they are inserted into Eq. (1) in the factor defined as follows:

(4)

Note that this may lead now in some cases to negative values of , the coefficient of contribution.

**Notes on effects from covariances:**

If covariances are considered for the calculation of uncertainties negative values may occur in the column “relat. contribut(%)“; this is not a program error.

According to the mentioned paper by Kessel et al. correlations (covariances) between input quantities are considered according to Eq. (1) in combination with Eq. (4) for calculating the “coefficient of contribution” and resulting values presented in the column “relat. contribut(%) in the uncertainty budget table. Eq. (3) then is no longer valid.

With UncertRadio this procedure according to Kessel et al. is implemented, i.e. the values shown in the column “relat. contribut(%)“ of the uncertainty budget correspond to this new definition. This may be demonstrated with the **example projects** **Kessel-2a-2006.txp and Kessel-2b-2006.txp**, which were prepared from two examples from that publication.

## 4.5 TAB “Results”

Under this TAB selected by mouse click the total result for the output quantity is shown including further variables and the values of the Decision threshold and the Detection limit. The output quantity is indicated which this result is referring to.

These are in detail:

***the result of the measurement:***

* the value of the output quantity
* the expanded uncertainty, in the same unit as that of the output quantity
* the relative expanded uncertainty (in %)
* the coverage factor (can be modified in the **menu Options**)

[***best estimates according to Bayes and confidence limits***](#URH_BestBayes_EN) ***(see also ISO 11929:2019):***

* the value of the output quantity
* the expanded uncertainty
* the value of the lower confidence limit
* the value of the upper confidence limit
* probability associated with the confidence interval

The toggle button „min. Coverage interval“ can be used to switch the display between probabilistically symmetric and the shortest coverage intervals, also in the case of the MC-Simulation.

***Decision threshold and Detection limit:***

* the value of the Decision threshold including the number of iterations (actually no iterations)
* the value of the Detection limit including the number of iterations
* the applied quantiles of the normal distribution, and k\_beta = , corresponding to the errors of first and second kind

***WLS, PLSQ, PMLE or WTLS: Standard uncertainties of the fitting parameter corresponding to the output quantity from the analysis of decay curve:***

* the uncertainty obtained from the least squares analysis; it is **NOT** multiplied with if the reduced Chi-squared value is larger than 1; this variant of the uncertainty of the net counting rate is used for estimating the uncertainty of the output quantity;
* that value of the uncertainty of the output quantity which is obtained from uncertainty propagation of the arguments of the Linfit function (i.e., mainly the background counting rate, if applicable with blank contribution) and of the uncertainties of the gross counting rates of the decay curve
* the value of the reduced Chi-square

A [***Monte Carlo Simulation***](#URH_MC_SIM_EN) may be started as a modern alternative to the propagation of uncertainties:

* input of the number *N* of simulated calculations of the output quantity (defining one run)
* input of the number of runs *r*
* Optional: selection of the coverage interval of shortest length (shortest **c**overage **i**nterval)

The MC simulation is started by clicking the button “Start”. The iteration number is indicated when iteratively estimating the detection limit.

From the r-fold repetition (runs) means and relative standard deviations (in %) are determined for:

*Best estimates according to Bayes:*

* the output quantity
* the expanded uncertainty
* the relative expanded uncertainty (%)
* the lower confidence limit
* the upper confidence limit

*and:*

* the Decision threshold
* the Detection limit

A **new Button "Save values"** was introduced. It can be used to transfer all the values being visible in the dialog, including those obtained by MC simulation, together with project name, date/time of execution into a CSV file: UR‑Saved-Results.csv. If this file does not yet exist, it is opened; then, records of data are appended to that file. The meaning of the columns is similar to those of the file AutoReport-Result.csv; however, there are further columns for the LINFIT parameters, for each of the output quantity, the decision threshold and the detection.

**In an extra dialog the three distributions are displayed which have been obtained from the Monte Carlo simulation.**

In the **menu Options** one can define the two values of the **normal Quantiles** corresponding to the errors of first and second kind, respectively. See also:

[**Implication of changing parameters within the Options menu**](#URH_AEND_OPTION_EN).

# 5 Running UR in batch mode

## 5.1 Using command line arguments

It is possible to run the UncertRadio program with adding command line arguments to the calling string. The command line arguments, abbreviated by Arg\_n, are strings (without blank characters in them!) which are passed over to UR as a sort of keywords at the starting time and may serve e.g. for passing over the project filename to be processed to UR. Such a call under Windows looks like:

>Start D:\Uncertradio\Uncertradio.exe Arg\_1 Arg\_2 Arg\_3

For a first step towards an automated type of calculation of several UR projects, one after another, the following three command line arguments have been implemented, which are interpreted by UR:

>Start D:\Uncertradio\Uncertradio.exe AUTO Inputfile.txp Sample\_ID

The first argument AUTO (capital letters) initiates the batch mode; the second argument is the name of the UR project file to be processed; the third argument is a sort of sample identification number (or a number within an experimental batch) and can e.g. be used to differentiate between such calculation results, which have between obtained for different samples but for different copies of the same master project file in which only a part of input parameters changed from experiment to experiment.

Another batch mode is implemented by which a given project can be evaluated serially for specified varied input values. The evaluation corresponds to that one described in chapter 5.6. The above argument AUTO is replaced by BATSER, followed by the project file and the CSV file with records of specified input values in it. The full command line then is (see chapter 5.6):

>Start D:\UR2\Uncertradio.exe BATSER J-ALUFT-Sr89-Sr-90\_EN.txp J\_Aluft\_serial\_EN.csv "LC=EN,;"

## 5.2 Batch mode processing with an Excel application

The automated usage of UR is of practical value if the UR calls are embedded in another program, which also takes control over the sequence of calls to UR with different project files. As an example, this may be a user-written program for processing the data evaluation for a certain measurement procedure, where a master project file is used for evaluating different batch samples such that only a part of the input parameters varies from measurement to measurement. The latter, however, requires that the user delivers a program part which modifies the “variable” input parameters in a copy of the master project file based on the [input format of a TXP project file](#_Structure_of_the). This work needs to be done by the user.

Based on the CSV format of an UR project, a first proposal for demonstrating the automated UR calculations has been established by Visual Basic within Excel. The filename of this Excel application is:

UR2\_SingleAutoRun\_V12.xlsm.

In this Excel file four spreadsheet tables are reserved for UR, which by default are set as Table4, Table5, Table6 and Table7:

Table4 : destination for import of an UR project as CSV file

Table5 : destination of the result records obtained from having executed UR

Table6 : list of UR project filenames for batch-like processing by UR

Table7 : contains three buttons for executing different VBA macros

The Excel file contains a module called Modul\_Auto\_Single\_UR. At the beginning of this module an if statement guarantees that 32 bit as well as 64 bit versions of Excel can be used. Thereafter, the first of the four spreadsheets mentioned above is defined:

Public Const FirstTableNum As Integer = 4

which may be changed by the user.

All other tables (spreadsheets) including newly inserted ones, except of these four, may be freely used.

The module Modul\_Auto\_Single\_UR containes several routines (macros) which are shortly detailed below.

|  |  |
| --- | --- |
| Init\_pathnames | In this small macro one has to fix the path-names of UR and this Excel file. ***The path-related information is taken directly from the file* UR2\_cfg.dat*, with one exception: the full path-name of the UR-Exee file must be inserted by the user in the cell A43 in Table7.***  Sub Init\_pathnames()  ‘Read other the filenames from UR’s configuraton file UR2\_cfg.dat:    **Fnum = FreeFile()**  **Reading the records of UR2\_cfg.dat depends on the format**  **of that file:**  **If the file is UNIX-formatted (record end LF), Excel**  **aggregates all individual records into a single one;**  **If the file is WINDOWS-formatted (record end CHR LF),**  **Excel extracts the array of all individual records.**  **Close #Fnum**    **‘ Language dependencies:**  ‘Set Decimalpoint and ListSeparator characters :  sDecimalPoint = GetDecimalSeparator()  sListSeparator = \_  Application.International(xlListSeparator)  ‘Set language:  **Win\_langg = "EN"**  **Select Case Application.International(**  **XlApplicationInternational.xlCountryCode)**  **Case 1: Win\_langg = "EN"**  **Case 33: Win\_langg = "FR"**  **Case 49: Win\_langg = "DE"**  **End Select**    ...  End Sub  It is assumed that the UR project files are located in a subfolder given by the variable example\_path. |
| Autorun\_UncertRadio | A simple macro that allows a batchlike processing of those UR projects, after they have been selected within **Table6**. It is invoked by a button from **Table7** (see below). |
| Import\_UR\_CSV\_file | This macro allows importing an external UR project file given in CSV format into **Table4** of the Excel file. It is invoked by a button within Table7 (see below).  This routine contains at its beginning an If-Then construct, which *by its activation* allows with „Run\_SheetName“ to select a name of the worksheet. |
| SingleRun\_UR | After editing of a project already existing in Table4, this macro exports it into a CSV file external to Excel, lets UR execute this project and finally imports corresponding result records into Table5. It is invoked by a button within Table7 (see below).  In detail:  export of the edited **Table4**: Makro DoTheExport,  execute this external CSV file with UR: Makro DoSingleRun\_UncertRadio,  Import the results obtained by UR to **Table5:** Makro doFileQuery. |
| Run\_UR\_AUTOSEP | This macro also calls SingleRun\_UR (with a new public variable UR\_AUTOSEP=True), but uses another two tables (sheets), UR2\_data und UR2\_results, for the project and the result values, respectively; UR2 in this case does not save data to the Auto\_Report file; at the end, two new CSV written by Excel and UR2 (with extensions \*\_xls.csv und \*\_xls\_res.csv) are deleted. |

Just between manually invoking the two macros Import\_UR\_CSV\_file and SingleRun\_UR is the time in which the input data contained in Table4 can be edited by the user, e.g. by entering new input data belonging to the next measurement evaluated by the same project.

After running of these two main macros the results (Table5) can be used for transferring them into other Excel files.

Within the VB code (makro Autorun\_UncertRadio) the total command string required for starting the evaluation of an external project, stored in the variable UR\_string, reads as follows:

UR\_string = Trim(UR\_path) & "uncertradio.exe AUTO " & Chr(34) & Trim(UR\_path) & \_

Trim(fname) & Chr(34) & " " & Trim(sid)

' add the language code LC=: (since 13.1.2018)

UR\_string = Trim(UR\_string) & " " & Chr(34) & "LC=" & Trim(Win\_langg) & \_

Trim(sDecimalPoint) & Trim(sListSeparator) & Chr(34)

Example for an UR string::

d:\UR2\uncertradio.exe AUTO "d:\UR2\zzURpr.csv" 556 "LC=DE,;"

The variables fname and sid contain the UR project filename and the Sample\_ID string. The pathname UR\_Path has to be fixed by the user at the beginning of the routine Autorun\_UncertRadio.

Within the VBA code of SingleRun\_UR the CSV project is transferred into that path which has been declared in the variable Excel-Path:

' write out the UR project CSV file:

file\_csv = Trim(UR\_path) & "zzURpr.csv"

Call DoTheExport(file\_csv, ifehl)

If (ifehl = 1) Then Exit Sub

' execute UR once with this input file:

Call DoSingleRun\_UncertRadio(file\_csv, ifehl)

If (ifehl = 1) Then Exit Sub

Processing the project file UR\_fname by UncertRadio is executed within Auturun\_UncertRadio with a function bShellAndWait. It causes Excel to wait until UR has finished its calculations and stopped. Then, within a loop, the next data evaluation is processed.

In the macro **DoSingleRun\_UncertRadio** the string holding the filename for the csv project output file has been changed (at two locations):

previous: file\_csv = Trim(UR\_path) & "zzURpr.csv"

since V. 2.4.04.: file\_csv = Trim(UR\_AUTO\_output\_path) & "zzURpr.csv"

previous: file\_csv = Trim(UR\_path) & filename\_org

since V. 2.4.04.: file\_csv = Trim(UR\_AUTO\_output\_path) & filename\_org

The four command line arguments of calling UR2 in the macro **DoSingleRun\_UncertRadio** are:

AUTO or AUTOSEP (%1)

trim(fname) (%2)

sid (%3)

LC=.. (%4)

The evaluation results obtained by UncertRadio for a project file are stored in a CSV file in a table-like structure. The name of the output file is fixed within UR:

CSV file: AutoReport-Results.csv

The output of data into this file is done in a cumulative form (appending rows at the end of the files). The numbers are written with using that decimal-point character which is defined within Windows.

These two files may be deleted if they have grown; UR then produces a new ones.

Meaning of the columns in the UR output files:

|  |  |  |
| --- | --- | --- |
| Spaltenbez. | Bedeutung | Meaning |
| # | Nummer der Ergebnisgröße | number of the output quantity |
| File | UR-Projekt-Dateiname | filename of UR project |
| Sample\_id | Proben/Analyse-Identifikation | identification of sample/analysis |
| Date | Datum + Uhrzeit | date and time of evaluation |
| quantity | Symbolname der Ergebnisgröße | name of the output quantity’s symbol |
| PE | Wert der Ergebnisgröße | value of the output quantity |
| uPE | erweiterte Unsicherheit, enthält den Faktor k, s. weiter unten | value of expanded uncertainty using the coverage factor k; see below |
| BE | bester Schätzwert | best estimate |
| uBE | dem besten Schätzwert beigeordnete erweiterte Unsicherheit | uncertainty associated with best estimate |
| LQ | untere Grenze des Vertrauensbereichs | lower limit of the confidence interval |
| UQ | obere Grenze des Vertrauensbereichs | upper limit of the confidence interval |
| sLQ | untere Grenze des kürzesten Vertrauensbereichs | lower limit of the shortest confidence interval |
| sUQ | obere Grenze des kürzesten Vertrauensbereichs | upper limit of the shortest confidence interval |
| DT\* | Erkennungsgrenze | decision threshold |
| DL# | Nachweisgrenze | detection limit |
| NT | (Nachweisgrenzentyp; sollte nur noch 1 sein, d.h. ISO 11929) | type of detection limit calculation (can only be 1, according to ISO 11929) |
| k | Erweiterungsfaktor für die Unsicherheit | coverage factor k for the uncertainty |
| kalpha | Wert von *k*1-α | value of *k*1-α |
| kbeta | Wert von *k*1-β | value of *k*1-β |
| 1-gamma | Wahrscheinlichkeit 1-γ für das Vertrauensintervall | confidence interval related probability |
| Chisqr | reduziertes Chi-Quadrat, im Falle linearer Entfaltung | reduced Chi-square value, in the case of linear unfolding |

## 5.3 Starting the macros

Start buttons have been implemented in **Table7** for invoking the three important macros:



## 5.4 Run UR with a project edited within Excel

After having imported an external UR-Project, given in the [CSV format](#URH_CSV_FORMAT_EN) ,



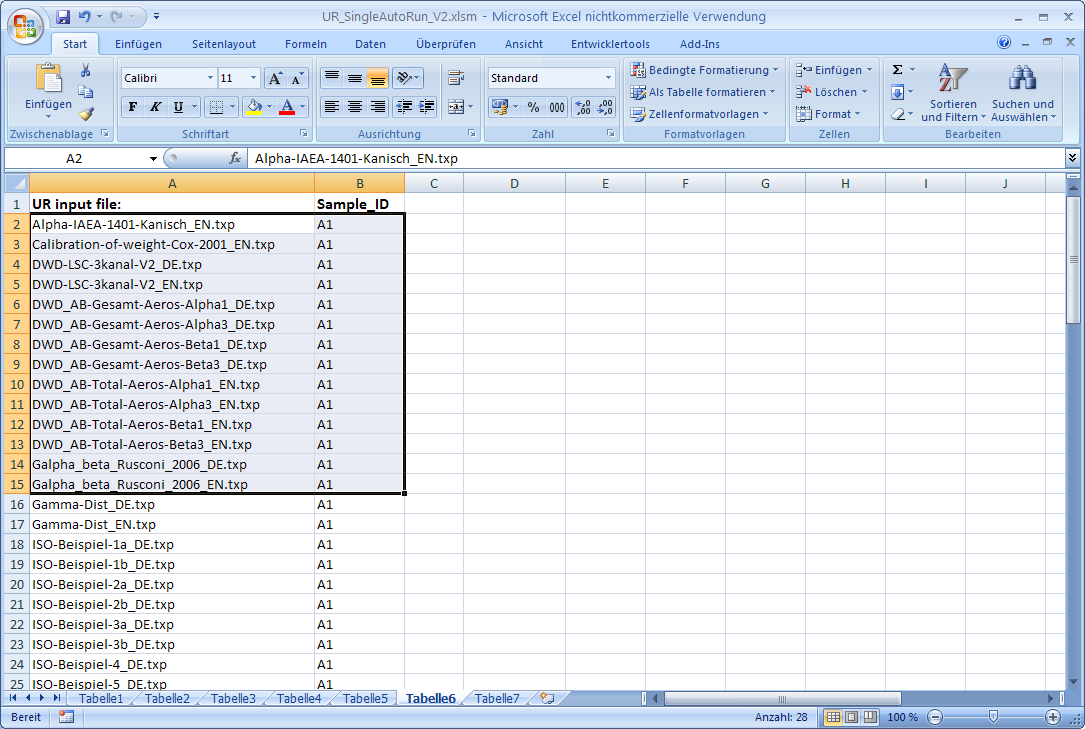
The data can be edited in Table4.

Then, the three steps of exporting Table4 into an external csv file, executing it with UR and re-importing the results record(s) can be invoked with a single Button in **Table7**:



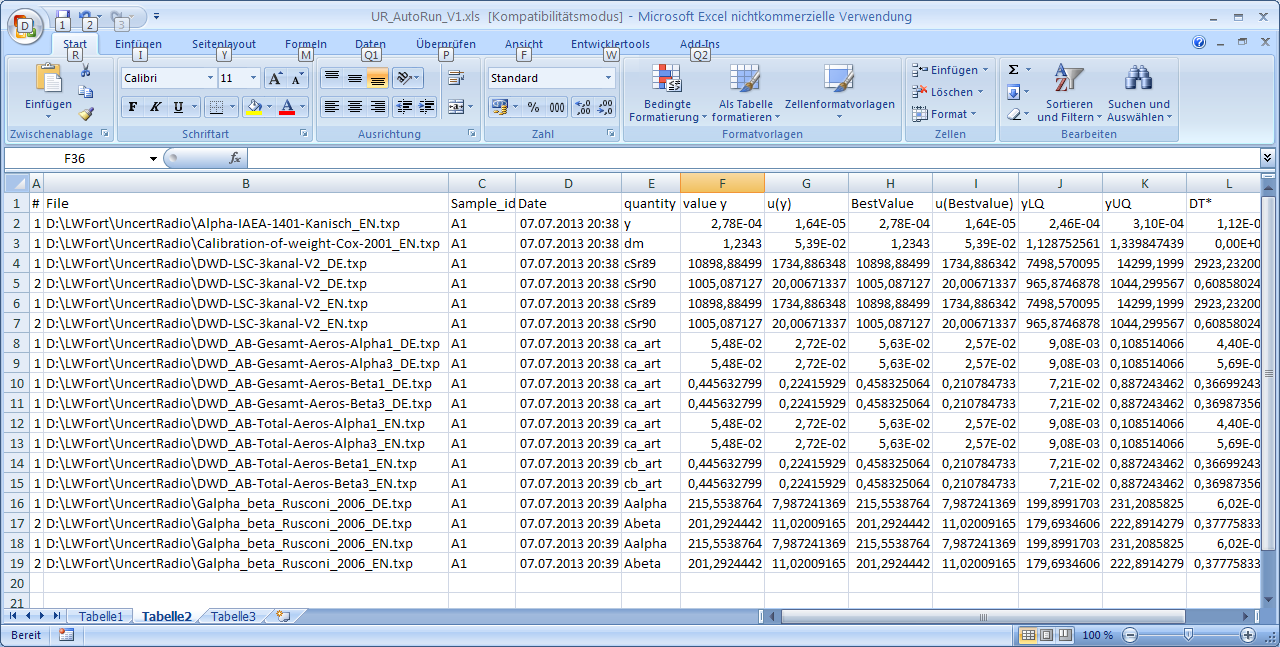
## 5.5 Example of an Excel application for a batch-like processing

The filenames of UR projects to be processed are collected in column A of Table6 and the associated Sample\_IDs in column B, respectively. In this example all UR example project files belonging to the program are listed there.



Before starting the batch processing with the button Start Autorun\_UncertRadio the associated filenames have to be selected as a two-column block as shown in the screen shot above.

After the batch processing loop in the VB code is finished, the CSV output file AutoReport-Result.csv obtained by UR is copied to Table5 at the end of that VB subroutine. A part of that sheet is shown in the following screen shot.



## Serial evaluations of an existing project

The necessity may arise to re-evaluate an existing project for the case of modified values/uncertainties of some of its input quantities. Modified values may result from further measurements; they may also occur, e.g., in an exercise of how the values of the decision threshold of the detection limit depend on some of the input values/uncertainties. Such a procedure is now supported by the menu item „**Edit – Serial evaluation**“.

Values/uncertainties of some of the input quantities to be modified can be transferred by the user into a CSV file, an example of which might be the following:

eps1; u(eps1); eps4; u(eps4)

0.338; 0.045; 0.390; 0.055

0.36; 0.045; 0.370; 0.049

0.35 0.047; 0.360; 0.052

The first line contains the symbols of the input quantities the values/uncertainties of which are to be modified; it is followed by lines (=sets) of modified values. Up to 60 symbols/values can be used; the symbols must be ones being already defined in the project. The first line is to be interpreted as follows:

Symbol the modified value associated with Symbol

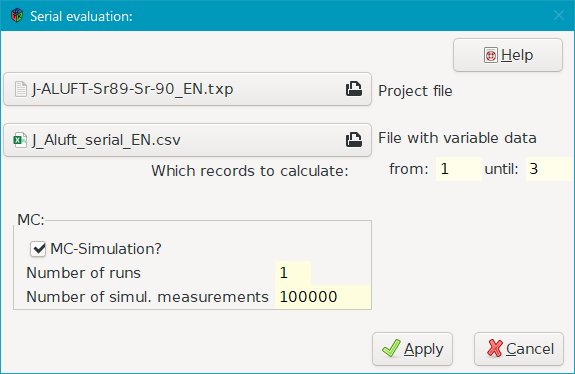
u(Symbol) the modified standard uncertainty associated with Symbol

hw(Symbol) the modified half-width value associated with the rectangularly distributed Symbol (from which the uncertainty is calculated internally)

Allowed symbols in this context are those being declared as “independent“ in the table “Values, Uncertainties“,

The example given above is now part of the UncertRadio installation as a file called J\_Aluft\_serial\_EN.csv. It is meant for using it with the existing project J-ALUFT-Sr89-Sr-90\_EN.txp.

With activating the menu item “**Edit – Serial evaluation**“, the following dialog is invoked by which the evaluation can be started after having defined the setup of this evaluation:



UncertRadio then produces one or two output files (csv type) for the results obtained – without and with MC simulation. Their names are derived from the name of the input csv file, as in the case of the above example:

J\_Aluft\_serial\_EN\_res.csv

J\_Aluft\_serial\_EN\_mc.csv

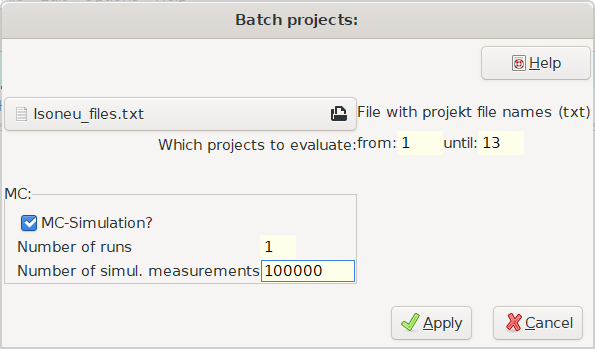
These files contain values for:

File; #EG; PE; uPE; BE; uBE; LQ; UQ; sLQ; sUQ; DT; DL;

(project filename, No. of the output quantity, primary value and uncertainty of the output quantity, best estimate and associated uncertainty, lower and upper quantile, decision threshold, detection limit)

## Processing projects in batch mode

UncertRadio allows to evaluate several projects in a batch mode. This mode is invoked by the menu item **Edit – Batch evaluation of projects** which opens a dialog, like that one shown in section 5.6. Therein, a simple text file is selected, which contains the project filenames, line by line.



UncertRadio then starts the calculations, with or without MC simulation, and writes the results into an **output file batch\_out.csv** (CSV format). For each evaluated project and each of its up to three output quantities, the filename is written, followed by a table of values for #EG, PE, uPE, BE, uBE, LQ, UQ, sLQ, sUQ, DT, DL. The column headers in the output file indicate their relation to the parts 1 and 2 of ISO 11929:2019.

Meaning of the symbols:

No. of the output quantity, primary value and uncertainty of the output quantity, best estimate and associated uncertainty, lower and upper quantile, lower and upper quantile with the shortest distance, decision threshold, detection limit.

In the case of MC simulation, a PNG graphics file is generated for each project and output quantity.

If MC simulation is selected, it may be recommended to use different batch list files, i.e., batch runs, for projects applying linear unfolding or not, because the MC simulation is much more time consuming for projects with linear unfolding.

## Batch-mode testing the evaluation of example projects

UncertRadio allows to evaluate all example projects in a batch mode. The results obtained are compared with reference results. Deviations found are documented in a file.

This test is invoked by the main menu item **Options – QC batch test**. The following dialog is shown:

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Two file names are pre-set which allow to directly start the test with the Apply button.

The first file contains the project filenames and the reference values of the example projects. The second file, the name of which may be modified, is the output file. In the case that deviations are found for a project, the reference values and the actually found values are given therein and compared for the individual characteristic values by ratios.

Due to previous modifications or corrections applied to 6 specific projects, their deviations from the reference values are documented for them. The only important point is whether deviations are found for others than these six projects. The deviations are given in detail in vgltest.txt.

This test takes about 47 seconds for a CPU rate with about 4 GHz. While running the test, only a static window is shown. The name of the project currently being processed is shown only in the headline of the UncertRadio window. After the test finished the following message is shown:

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Of the three text lines in this message, only the middle one is the relevant one. Only if more than zero deviations are given therein, this indicates a problem.

**This test is always applied before publishing a new version of the UncertRadio software**.

# 6 Special Methods

## 6.1 Uncertainty propagation

### 6.1.1 Methods without linear unfolding

An equation, or a system of several equations describing the evaluation of the measurement quantity, is called **evaluation model**. It is stated in the form

(1)

Inserting estimates for the input quantities yields an estimate of the output quantity, the **primary measurement result** ***y***:

(2)

The standard uncertainty associated with the primary measurement result is calculated according to the following equation, assuming that measured values of the input quantities are statistically independent:

. (3)

This is known as **uncertainty propagation**. The partial derivatives are also known as **sensitivity coefficients**. It is often written such that *G* is replaced by *Y*:

If the input quantities have been measured in a way that they are not statistically independent, associated covariances have to be taken into account. Then, the extended previous equation reads:

(4)

Here, represents a more general way of stating a covariance between two measured input quantities, which often are also given as or , respectively. The following holds: and .

Equation (3) is the basic form of the uncertainty propagation used in UR for methods not requiring linear unfolding. This is done by using the values and standard uncertainties defined under the TAB “Values, Uncertainties“. In case that covariances have also been declared in this TAB, Equation (4) is applied.

### 6.1.2 Extension to several output quantities

Equation (4) above can also be written as follows:

. (5)

Furthermore, by extending this to the case of linear unfolding for more than one output quantity (y then gets indices and ) this equation becomes:

, (6)

There exists an equivalent of this equation in matrix algebra notation, which indeed is applied in UR. It assumes an *n*-Vector and an *m*-Vector **Y**, associated with an *n*×*n* covariance matrix and an *m*×*m* covariance matrix , respectively. Introducing furthermore an *m*×*n* matrix with elements , i.e., partial derivatives,

, (7)

allows writing Eq. (6) as follows:

. (8)

This equation represents the way of doing uncertainty propagation, which is used especially when partial derivatives can be calculated numerically. Note that this equation is one which can, e.g., well be used within Excel.

Now consider the simple case, that *m*=1 and *n*=2 holds. This may occur when doing linear regression with 1 equation and 2 unknowns, where is the covariance matrix of the two fitted parameters and . The associated equation is . Then, Eq. (8) becomes (note that an uncertainty of would require an additional propagation term):

From this one recognizes the equation (4) being separated into single terms.

### 6.1.3 Applications in UR

The case of linear unfolding is represented by several equations of the type

,

or, in matrix notation (**A** contains the terms):

(9)

Now, the desired output quantities **Y** are found in the right side of this equation, the input quantities on the left side. As it is well known, the linear least squares-method yields the solution to this problem:

; (10a,b)

In a next step, it is assumed that the elements of the design matrix **A**, which e.g. consist of decay corrections when analyzing a decay curve, may contain further parameters with associated uncertainties (given as a vector **p** with covariance matrix ). Using a transformation matrix **Q** containing numerically calculated partial derivatives , an extended version of the covariance matrix can be calculated (this does not change the value of **y**)**:**

(11)

For every combination (with ) of numerical approximations of the partial derivatives

, (12)

the matrix **A**, the output vector **y** (its elements being considered as functions) and the covariance matrix must be recalculated.

Alternatively, by using the chain rule for derivatives the second term in Eq. (11) can be expressed as follows:

Herein are: : matrix of partial derivatives of *y*i with respect to the elements of the matrix **A**, **Dp** : matrix of partial derivatives of the elements of **A** with respect to parameters **p**; **UA** : covariance matrix of the elements of **A**, which are functions of **p**. The chain rule for partial derivatives takes the form **Cp**= **CA·Dp**, with the property  **.**

Especially the expression is required within the WTLS fitting procedure; it is derived in a subroutine which is used by both fitting methods, WLS and WTLS, respectively. The matrix **UA** is calculated as indicated in the preceding sub-chapter.

**Note on the covariance matrix UA:**

For linear unfolding with the WTLS procedure a test had been implemented with the previous version, which by using the Cholesky decomposition tests whether the input covariance matrix is positive definite. This has been improved by testing in advance of the Cholesky decomposition the Cauchy-Schwarz inequality:

.

For pairs (i, k), for which equality is found, the associated cov(i, k) is multiplied by the factor (1-δ) with δ=1·10-09. Should the covariance matrix after these tests again be not positive definite (it cannot be inverted) all non-diagonal elements are multiplied with this factor. Should this not help, the program evaluation is interrupted with giving a hint on this problem.

This primary result (i.e., the values of the fit parameters; they often represent activity values referred to the time of measurement) of linear may need further treatment, if the fit parameters values obtained must be inserted into to further equations. This is the case, if the activity value at measurement (fit parameter) needs to be divided by mass/volume and requires and a further decay correction and these new parameters (parameter vector **q**) associated with these corrections have uncertainties (**embedding linear unfolding**). This often may not require matrix algebra, if the corrections of the fit parameters consist in simply multiplying them with factors which results in the “final” output quantities :

(13)

Nevertheless, these functions may be arranged into a vector

. (14)

Calculating the covariance matrix of now requires, by analogy to Eq. (11), an (*n* x *n*)-matrix of partial derivatives of the functions with respect to the fitted values . For taking into account the uncertainties of the *n*q values **q** requires to calculate the correspondent (*n* x *n*q)-matrix of partial derivatives .

If there are no covariances between **y** and **q**, the extended version of Eq. (11) yields the covariance associated with the finally desired output quantities :

, (15)

where the covariance matrix **Uy** is the one from (11), which then results in:

(16)

Note: If the partial derivatives being the elements of the matrix **Q** (Eq. 11) are referred to the vector , i.e. replaced by , the term in Eq. (16) associated with them, hast to be removed from Eq. (16) and moved to Eq. (15) as an additional term.

For further reading, see:[*Cox et al., 2004*](#URH_LITERATUR_EN)*.*

## 6.2 Best estimates according to Bayes and confidence limits

The value ***y*** and its combined standard uncertainty ***u(y)*** obtained from the evaluation of the output quantity define the mean and the standard deviation of a Gaussian distribution of possible values

which is obtained from the application of the maximum information entropy principle.

Dependent on the size of the ratio ***u(y)/y*** it may happen that a part of this distribution resides in the negative domain (*< 0*). According to the Bayesian method it follows by using a positive “model prior“, that this distribution function shall only have positive values for positive values (: e.g. activity values). Therefore, the part of the distribution in the negative domain is cut at *=* 0. Using this modified (cut) distribution **expectation values of mean and standard deviation** can be determined which are here called “**best estimates according to Bayes**”. Let ω be the value of the integral of the modified distribution function from zero to infinity, with . The resulting expectation values then are:

mean:

with

( is the (cumulative) distribution function of the standardized normal distribution)

standard deviation:

This guarantees that always will have a positive value.

**The probabilistic symmetric coverage interval**

For the result value and the standard uncertainty , this interval is given by the two limits:

with

with

A normal distribution is assumed therein which is cut at the left side at the value of zero.

**The shortest coverage interval**

This interval, no longer be symmetric, is defined by the limits:

with

These are modified In the case of :

with

**Numerical estimation of these interval limits in the case of a MC-Simulation**

From a MC-Simulation of a physical quantity, an array of about 104 through 106 simulated values results which is sorted ascending. A simple relation between a value and the associated probability value holds:

with

For a given probability its associated index within the array is found by:

.

The associated value of the interval limit, considered as a quantile lies between the two adjacent values und .

The shortest coverage interval (min\_length) corresponding to the probability is searched for within a simple loop over the array y(i) as follows:

imax = \*N

min\_length = 1E+20

do i=1,imax

q\_left = y(i)

q\_right = y(int(N\*(1 – i/N)))

if(q\_right – q\_left < min\_length) then

min\_length = q\_right – q\_left

q\_left\_min = q\_left

q\_right\_min = q\_right

endif

enddo

## 6.3 Linear Least squares method

**Principle of the multi-linear least squares fitting**

**Model**

For the use of the (multi-) linear least squares fitting (LSQ analysis) the following model of a decay curve is assumed:

(1)

A sum of up to three terms being dependent on the counting time can be fitted to measured values of the quantity *Y* being also dependent on ; *k* counts the measurements. The are the coefficients, which are to be determined by fitting (fitting parameters). The quantities are functions (decay curves) which are considered to be known and which depend only on and other parameters, e.g., half-lives, which are not considered as fit parameters. They must not be dependent on the fitting parameters . is treated as dependent variable while the functions are treated as independent variables in the LSQ analysis. For the moment it is assumed here that the measurement of the values of is done with a single-channel counter.

By the introduction of up to three output quantities to be treated simultaneously, it is, for instance, possible that the simultaneous measurement of several radionuclides is done by using a two- or three-channel counter. Using an LSC counter with three counting channels (energy regions) allows for instance to determine simultaneously the activities of Sr-90, Sr-89 and Sr-85 (yield tracer).

If more than one output quantity is defined for a project, e.g., two or three representing the activities of different radionuclides, the program inserts three new symbols, Fitp1, Fitp2 and Fitp3, into the list of symbols, which refer to the values of the . These names are not allowed to be changed. From these, the user can derive with further equations (in the text window under the TAB “Equations”) above the *Linfit* call the decay corrected activity concentrations. Under the TAB “Values, uncertainties” standard uncertainties obtained from the linear curve fitting are assigned to these new symbols. Additionally, their possible correlation pairs and associated correlation coefficients are inserted into the corresponding table under the same TAB which enables their whole covariance matrix of the three fitting parameters to be used for the uncertainty propagation of the output quantities.

Note, that also in the case of only two output quantities all three symbols, Fitp1, Fitp2 and Fitp3, are inserted into the symbol list. Only, **if only one output quantity** is defined, **none** of these parameters is inserted in the symbol list. In this case the value of the fitting parameter is attributed directly the quantity *Rn* in the call *Rn = Linfit(1,… )*; c.f. further down.

**Least squares routines used In UncertRadio**

Two different routines are used in the program for least squares estimation. These are:

* **the “simple“ least squares procedure (LSQ) which is usually used if the values do not have uncertainties**; the measured values , however, have uncertainties, covariances between them are also taken into account. If the values nevertheless have uncertainties, they are included by UncertRadio within the uncertainty propagation outside the LLSQ routine;
* In autumn 2013, three options for selecting a fitting procedure were introduced, which differ in their associated Chi-square expressions:
* **WLS**: Using the **Neyman Chi-square**; this procedure is identical to the previous procedure NLSQ; linear, without iterations;

**PLSQ**: Using the **Pearson Chi-square**; linear / iteratively;

**PMLE**: Poisson Maximum Likelihood Estimation (Poisson MLE); non-linear / iteratively.

* **the “general case“ of least squares (WTLS, weighted total least squares)**, **which in addition considers uncertainties of the and possible covariances between them.** Herein, an iterative, i.e., non-linear, matrix procedure is used, which is a more time-consuming method because of the iterations. The possible covariances between the values are determined by the program internally by applying partial derivatives with respect to the symbols contained in the functions; they need not to be supplied by the user.

For the background information of these fitting methods see [chapter 7.4.3](#URH_Chisq_Options_EN).

By default, the “simpler” WLS procedure is invoked by the call *Rn = Linfit(1,… )*. The use of the WTLS procedure may be selected within the dialog Definition of the decay-curve model.

**Notes:**

**Correlations between measured values may occur.** This for instance is the case if the measured values are net counting rates for which the same value of a background counting rate as well as the net blank counting rate have been subtracted from the corresponding gross counting rates. These covariances - they can be calculated internally from a formula - need to be considered especially in the case of quite low net counting rates.

In the case of the net counting rates () the covariance formula between two of its values is given by:

.

This is internally used in the program to derive the complete covariance matrix to be used by the fitting routine. In this equation, must be set equal to zero if either and are different or their uncertainties.

It has to be mentioned here that for this method the background counting rate can be defined in the special [**input dialog for decay curves**](#URH_EINGABE_ABKLINGKURVE_EN). For each individual gross counting rate of the curve an individual background counting rate may be given – this may occur with LSC measurements –, or one single value of the background counting rate () may be used for the whole curve. must contain only the detector-related background component.

Furthermore, a counting rate is required which refers to a chemical blank analysis where the detector background component is already subtracted. Therefore, is considered as a „net blank count rate“; it quantifies the background due to chemicals and glassware used during an analysis. The symbol *Rbl*, with value and uncertainty, is part of the table under the TAB “Values, uncertainties”, the values of which are transferred by the Linfit Call (c.f. further down in this theme) to this numeric routine.

For the mathematics see:

1. [**linear curve fitting with WLS**](#URH_LLSQ_MATHE_EN)**,**
2. [**linear curve fitting with WTLS**](#URH_GLSQ_MATHE_EN)**.**

Definition of the model: see [***Dialog Definition of the decay-curve model***](#URH_MODELL_FESTL_ABKLG_EN)

After having defined the Linfit-Call within the equations, the use of the WTLS procedure may be selected in this dialog.

**Activating the (multi-) linear Least squares fitting**

This is shown for two examples of different complexity.

1. **Simple example:**

*Assumtion*:

Number of output quantities: 1; from the LSQ fitting a net counting rate *Rn* is obtained.

Within the text field for equations at the location where otherwise the net counting rate *Rn* is defined, for instance

*Rn = Rg – R0 ,*

this equation is replaced by the following:

*Rn = Linfit(1, Rbl, HwzY90, Hwzlong, HwzAc228, tmess, tstart)*

**Linfit** is the name of the procedure which initiates the LSQ fitting with its associated sub-dialogs. Its parameters are:

*1* No. of the variant of this measurement evaluation task for which this type of fitting shall be used; at present not more than the present variant of the evaluation of an Y-90 decay curve analysis exists;

*R0* background counting rate including also blank contributions which is subtracted from the measured Y-90 gross counting rates, in ;

*t0* counting time of the background measurement, in ;

*HwzY90* half-live of Y-90, in

*Hwzlong* half-live of a longer-lived radionuclide contributing to a (slowly decaying) background, in ; e.g. Th-234; if Hwzlong = 0 is set the associated decay factor is set internally equal to 1

*HwzAc228* half-live of the possibly interfering radionuclide Ac-228, in *;* this cal also simulated a contribution of short-lived radon decay products

*tmess* place holder for the counting times of the individual counting times belonging to the net counting rates

*tstart* place holder for the periods of time between the time of the Y-90/Sr-90 separation and the starting time of the individual measurements

**Note:** Only recently, **UncertRadio was modified such that only the three parameters *Rbl, tmess* and *tstart* need to be given** in the Linfit call: *Rn = Linfit(1, Rbl, tmess, tstart)*

After loading the symbols from the equations including that Linfit-call described above the symbols from this routine are available in the common list of symbols. In the TAB “Values, uncertainties” values and uncertainties of the symbols *R0, t0, HwzY90, Hwzlong* and *HwzAc228* have to be entered then, however, not for *tmess* and *tstart*.

With one exception, of course, one may use other Linfit symbols instead of those shown above, they only need to be given in the total symbol list; these symbols are to be considered “globally” valid.

Important: Only the symbol names *Rbl*, *tmess* and *tstart* must not be changed, which is also true for their meaning as defined above.

After the call to Linfit, the value of the fitting parameter and its uncertainty have been transferred to those of the symbol *Rn*.

1. **More complex example:**

*Assumtion*:

Number of output quantities: 3; as a result from the LSQ fitting one obtains the fitting parameters Fitp1, Fitp2 and Fitp3, corresponding to the parameters , which now represent the activities (in Bq) of Sr‑89, Sr-90 and Sr-85. This is an example taken from the example project DWD-LSC-3kanal-V2.txp.

Within the text field for equations at the location where otherwise the net counting rate *Rn* is defined, for instance

*Rn = Rg – R0,*

this equation is replaced by the following:

*rd = Linfit(1, Rbl, eSr85A, eSr85B, eSr85C, eSr90A, eSr90B, eSr90C, eSr89A, eSr89B, &*

*eSr89C, eY90A, eY90B, eY90C, lamSr85, lamSr90, lamSr89, lamY90, tmess, tstart )*

***This call can be shortened to****:* *Rn = Linfit(1, Rbl, tmess, tstart).*

The meaning of the symbols is equivalent to those in the „simple example” given above. The symbol names *Rbl*, *tmess* and *tstart* as well as their associated meaning must not be changed. The symbols *eNuklidX* (in total 9) designate detection efficiencies of the different radionuclides for the counting channels A, B or C. The ssymbols *lamNuklid* represent the decay constants of the three radionuclides.

Note:

Apart from the fixed symbols *Rbl*, *tmess* and *tstart,* which have to appear in the Linfit call, neither the names of other symbols are fixed nor their number; however, they must appear in the whole (global) symbol list. They must be used in the equations defining the functions ; for more information, see also [***Dialog Definition of the decay-curve model***](#URH_MODELL_FESTL_ABKLG_EN).

For Input of data from the decay curve see: [**Dialog “Values of decay curve”**](#URH_EINGABE_ABKLINGKURVE_EN)

[**Viewing the result of the decay curve LSQ fitting**](#URH_LSQ_ERGEBNIS_EN)

[**Note on the procedure for calculating Decision threshold and Detection limit in the case of Least Squares fitting**](#URH_LSQ_NWG_EN)

## 6.4 Utilizing a calibration curve

There were demands existing to consider within an evaluation a specific input quantity (e.g. the detection efficiency) as being dependent on “another” quantity (such as area mass density, or a quench factor), which means to take its value from an associated “calibration curve” by interpolating a polynomial function of the “other” quantity.

For this purpose, a new function KALFIT was established, for which a new dialog allows input of values x(i) and y(i) together with their standard uncertainties. x and y (independent and dependent variables, respectively) represent measured values of a calibration curve. The curve is modeled by a polynomial with maximum degree of 3 (max 4 coefficients). The polynomial coefficients are calculated by a weighted or non-weighted (multi-) linear least squares fit.

(1)

Unused (empty) columns in this dialog’s grid are set internally equal to 1.

Choosing a polynomial degree of 0 (i.e. 1 coefficient to be fitted), fitting of the *y*-values results in a **weighted mean (uncertainties given)** or in a **non-weighted mean (uncertainties not given).** Its standard uncertainty is that of the mean, i.e. already divided by the square root of the number of values.

**Activating the calibration curve tool:**

eps = KALFIT(1, eskv) (2)

In this example KALFIT is called to determine the detection efficiency value **eps** (and its uncertainty) **as a function of** a quench factor **eskv** by using the fitted polynomial for interpolation.

The second parameter (i.er., eskv) represents a value of the **independent quantity X**, by which value and standard uncertainty of the dependent quantity Y (in this case of **eps**) are determined.

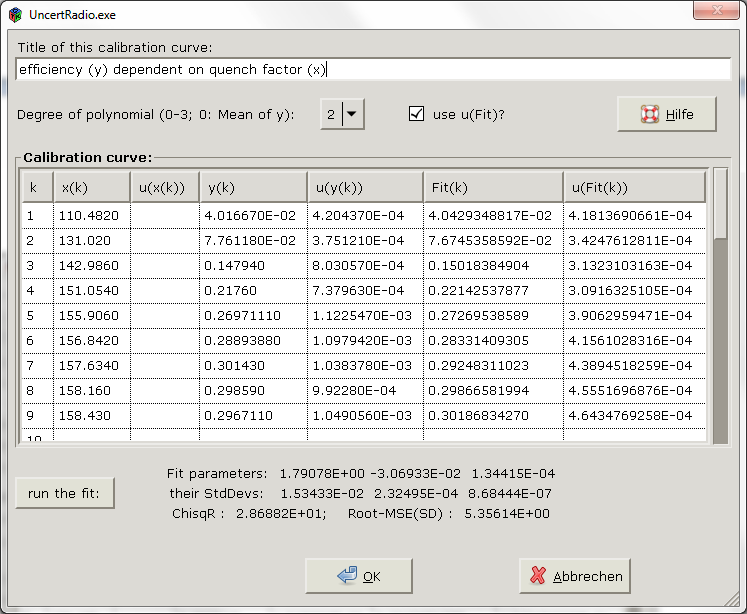
The first parameter of this function (here: 1) gives the information about how to use the calibration curve for calculating the value of the left side of Eq. (2). The value 1 means that the value for Y is calculated (read) as polynomial just as shown above. The value 2 means that the value for Y is determined by reversing the polynomial. An example for the latter case is treated in the UR project Example\_8\_with\_KALFIT\_EN.txp), for which the Eq. (2) above is replaced by:

Cx = KALFIT(2, Rnet)

It means that Rnet designates count rates, which in a first step are calibrated as polynomial function dependent on known concentration values x: Rnet = Polynomial(x). The second step is to measure a count rate Rnetx of another sample and to determine then its associated concentration Cx. Assuming a degree 1 of the polynomial one would calculate the concentration Cx by (Rnetx-a1)/a2;for a higher-degree polynomial a numerical bisection method is used for the inversion. In the mentioned UR project, the concentration refers to that of Potassium given in g/L.

This KALFIT call invokes a new dialog (there is also a new item under the main menu Edit), by which the x- and y-values of such a calibration curve (including their standard uncertainties) can be input; the desired value for eps is taken via its x-value eskv from the polynomial curve for the y-values. Another button allows executing the polynomial fitting (max. degree of polynomial = 3) such that the final value and the standard uncertainty are made available to UR for the quantity eps.

The standard uncertainty of the desired Y-value (eps) is calculated by numerical uncertainty propagation using the fitted parameters and their covariance matrix and the uncertainty of the given x-value (eskv).

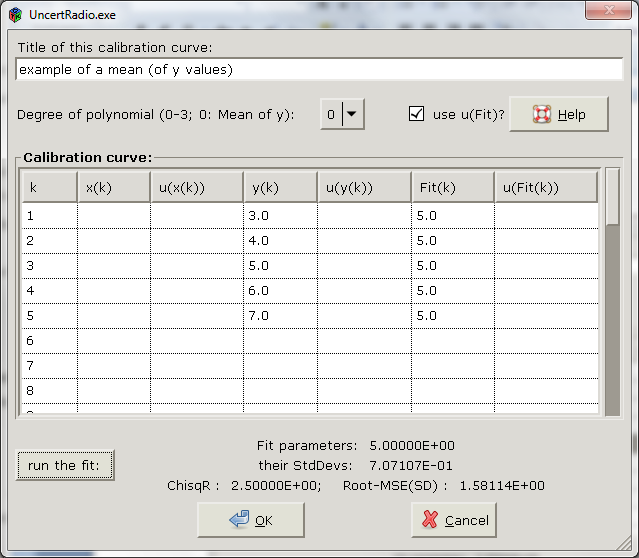


Although eps – as defined by the equation above – formally is a dependent quantity, it is treated in UR as if it were an independent quantity, e.g. regarding the uncertainty budget.

Currently, UR assumes that the value of only one input quantity is determined by a call to KALFIT.

The uncertainties of the x-values are currently not considered.

The following dialog shows an example of calculating a mean value (degree of polynomial equal to zero); the values to averaged are always the ones in the column for y(i).



## 6.5 Activity determination from several gamma lines

In this example of use from the **gamma-ray spectrometry, preferentially with high-resolution Germanium detectors**, it is assumed that some different radionuclides may occur in the measured source, but the gamma lines (more than one) of that radionuclide for which the activity shall be determined must not interfere with other lines in the spectrum. The case that the activity shall be determined from only a single line, is not considered here, because that can be done with the standard procedure of UncertRadio.

Let the net counting rates of n gamma lines of the radionuclide be given. From these the activity values , in Bq, are calculated with the following equation:

(1)

The symbols herein mean:

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbols:** | **Meaning:** |  | **In Windows Dialogs**  **written as:** |
|  | net counting rate of the gamma line at energy , in |  | RnetRate or  PeakNetRate |
|  | full-energy peak efficiency at energy |  | effi, eps or epsPeak |
|  | (absolute) emission probability of the line i |  | pgamm |
|  | self-attenuation correction at energy |  | fatt |
|  | correction for coincidence summing of a line at energy |  | fcoinsu |

The standard uncertainties of the activities of single lines calculated according to Eq. (1) are calculated internally in UncertRadio by using the uncertainty propagation for Eq. (1).

**Note:** Covariances between the calculated activities are considered. Such covariances e.g. may be inferred by reading efficiency values from the same FEP efficiency curve , because their values for different energies are derived from the same identical set of parameters obtained from fitting the curve. Considering this would require the inclusion of the full covariance matrix of these parameters, which however cannot be handled by the program. Instead, single values of the covariance/correlation can be input under the TAB “Values, Uncertainties”.

**Description of procedures for calculating a mean**

**a)** [**Weighted mean**](#URH_GSPK1_WMEAN_EN)

**b)** [**weighted least-squares mean**](#URH_GSPK1_LSQ_WMEAN_EN)

[**Procedure for calculating Decision threshold and Detection limit for Gamspk1**](#URH_GSPK1_NWG_EN)

**Invoking the evaluation of several gamma lines**

In the text field for equations the following call is used in that place where one otherwise would define the activity *A* of the counting source:

*A = Gamspk1(E, tlive)*

**Gamspk1** is the name of the procedure being activated with its sub-dialogs which is doing the calculations of the chosen mean. Its parameters are:

*E* placeholder for the energies of individual gamma lines, in keV; the program automatically attributes values to

*tlive* counting time (duration) (live-time) , in ;

After loading the symbols from the equations they are available to the **Gamspk1** function. Within the TAB “Values, uncertainties” a value and its uncertainty must be given for the symbol *tlive*, but not for *E*. Of course, instead of *E* and *tlive* other symbol names can be used, they only must be defined in the symbol list of the project. These symbols are defined always as “global” variables.

Note: For the source activity *A* as defined here the decay correction and e.g. a mass or a volume have to be defined in the equations outside of Gamspk1.

**Input of values:**

for further information about input: [**Dialog Values from spectrum evaluation**](#URH_GSPK1_WERTE_EN)

[**Viewing the result of mean calculation with Gamspk1**](#URH_GSPK1_ANSICHT_EN)

## 6.6 Monte Carlo Simulation

Under the TAB “Results” one can start a **Monte Carlo simulation** for cross-checking the value and the uncertainty of the output quantity ***y***.

The simulation is done for a chosen (large) number of simulations of the measurement. For this purpose, for each of those quantities having been defined in the symbol list under the TAB “Equations” and characterized there as independent (u) input quantities, simulated input values are taken from their correspondent distributions (normal, rectangular or triangular distributions). The underlying individual distributions have been determined under the TAB “Values, uncertainties”. If the [**(N+x) rule**](#URH_NP1REGEL_EN) has been selected there for counting numbers, their associated values are MC sampled according to a Gamma distribution; the values of the counting rates derived from the counting numbers then are also Gamma distributed. If all these input quantities have got a simulated value, a value of the output quantity - “the first simulated value of ***y***” - is calculated according to the equations already defined.

From the many-fold repetition of this step a statistical distribution of the values of the output quantity is obtained from which its best estimate and its associated standard uncertainty are calculated as arithmetic mean and standard deviation, respectively. At present, with this method only quantities having normal, rectangular or triangular distributions can be considered.

[**Obtaining MC distributions and statistics derived of it in detail**](#URH_MCDETAIL_EN)

The great advantage of this method is that partial derivatives with respect to the independent quantities are not needed!

**Note**: the random generator used here has a period of about .

The procedure just described is correct for the case that no correlations exist between the input quantities. If in the TAB “Values, uncertainties”, however, covariances between **pairs of correlating quantities** have been given, correlated simulated values must be attributed to these quantities.

**Important note on this issue**: This method of producing correlated variables was completely adapted to methods described in textbooks where matrix methods are used (keyword: “Gaussian distributed random numbers in n dimensions”; S. Brandt, Datenanalyse; V. Blobel and E. Lohrmann, Statistische und numerische Methoden der Datenanalyse; as well as in the Supplement 1 of ISO GUM). This can be proven by trying the UncertRadio example project files **Kessel-2a-2006.txp and Kessel-2b-2006.txp** from the recent publication by Kessel et al. (2006) (see also: [**Meaning of the TAB “Uncertainty budget”**](#URH_TABBudget_EN)).

Intermediate results of the MC simulation, partially consisting of tables, are collected in a separate text file MC\_Tables.txt.

Generally, one will find a good agreement between the results from the MC simulation method and from the analytical method. **Therefore, the MC method is a relatively easy and elegant alternative to the more extensive analytical procedure**.

**What can be followed from deviations between the two methods?**

If by using the MC method a result is obtained which deviates from that by the analytical method, one could easily conclude that there could be an error somewhere in the analytical procedure. However, this conclusion not always needs to be true!

**What the described MC method actually is calculating can be interpreted as “Propagation of distributions”.** This means that it in principle the expectation value of the output quantity is estimated, i.e. the following *n*-fold integral:

.

Herein, designates in compact form the equations which are necessary for the calculation of ***y*** (i.e. the formula with which the value of ***y*** is usually calculated) and the probability density functions of the *n* input quantities characterized as independent. The widths of the are determined from their associated measurement uncertainties . An important assumption for the conventional propagation of uncertainties is that the uncertainties of the should be small. In this case, the probability density functions approximately become delta functions with the consequence that the *n*-fold integral reduces to the conventionally calculated value of the output quantity ***y***. In this sense deviations between both methods may occur with respect to the output quantity and its uncertainty if any of the involved uncertainties “are not small” which might also be if such quantities belong to the denominator of the evaluation equation (non-linearity).

The collection of project files contains **examples in which the discussed deviation between both methods is significant**:

|  |  |
| --- | --- |
| **Project file** | **Special feature** |
| ISO-Example-1a\_EN.txp,  ISO-Example-1b\_EN.txp | Here, the alpha self-absorption factor *f*, having a rather broad rectangular distribution and belonging to the denominator, causes this effect. |
| Neutron-Dose-Cox-2006\_EN.txp | The field specific correction K with a significantly broad rectangular distribution has practically the same effect as in the wipe test example above: the resultant distribution is significantly asymmetric. |
| Calibration-of-weight-Cox-2001\_EN.txp | In this example a significantly larger measurement uncertainty results from the MC method. Rectangular distributions are attributed to three of the involved input quantities. |
| Wuebbeler-Ex1\_EN.txp | A non-linear model function in combination with large uncertainties of normal distributed input quantities result in an asymmetric distribution of the output quantity. |
| Wuebbeler-Ex2\_EN.txp | Rectangular distributed input quantities result in a trapezoidal distribution of the output quantity. |

## 6.7 Low-Level Applications, (N+x) Rule

**Important note: It turned out (2024) that the way of applying the (N+x)-rule as described below is not yet clearly stated in ISO 11929. Therefore, applying it or not is up to the user.**

Very low numbers of counts N are Poisson-distributed instead of Gaussian-distributed. Using Bayesian methods it can be shown that counting rates derived from them then are Gamma-distributed. Taking a prior assumed as uniform (Weise et al., 2009) results in an expected value of the counting rate which is characterized by with a value of for its variance. From this the recommendation follows, for the case of very low counting numbers, to replace by, **being designated as the Rule here** which is long known from the literature.

The symbol x in (N+x) may be considered as a variable designated as GamDistAdd in UR. This results in a Gamma distribution of the variable (count number) associated with this rule, which from a Bayesian view results from a prior which is proportional to 1/ρ (GamDistAdd=0.0), to ρ-1/2 (GamDistAdd=0.5) or which is constant (GamDistAdd=1).

|  |  |  |
| --- | --- | --- |
| *c*=GamDistAdd | (1-c) | mean |
| 0 | 1 | N+0 |
| 1/2 | 1/2 | N+1/2 |
| 1 | 0 | N+1 |

It is assumed here that this rule only refers to the gross counting rate *R*g and to the background counting rate *R*0, because other types of counting rates usually are determined by other methods being less direct than a measurement. Within the UncertRadio dialog, the values of these two counting rates do not require modification, if the Rule is applied directly to the gross and background counting numbers, *N*b and *N*0, respectively. Within the program code the following replacement rule is applied:

Directly before a program part which is doing calculations with the counting numbers:

*N*b is replaced by the term (*N*b+ GamDistAdd),

*N*0 is replaced by the term (*N*0+ GamDistAdd).

After that program part: these replacements are removed.

The default value of GamDistAdd is 0.

The application of the (*N*+x) rule to the two mentioned counting numbers is made available within the TAB “Values, Uncertainties” by selecting “(*N*+x) rule” as type of distribution.

**Note: Only for counting number variables the (*N*+x) rule may be selected; it must not be selected for the associated counting rate variables.** The latter are treated in this way internally and thereby are also gamma-distributed. This means, that this rule is correctly applicable only if a counting rate R is defined by an equation R=N/t containing the associated number of counts N. For N only a value is given by the user, while the field for the uncertainty must be left empty; this uncertainty obtained by the gamma-distribution is then calculated internally.

Two example projects (Gamma-Dist\_EN.txp and Lira\_gammdist\_EN.txp) demonstrate the application of the (*N*+x) rule for the case of very low counting numbers.

**Important change: In anticipation of the new version of ISO 11929:2019 the application of the (N+x) rule has been modified as follows. Apart from a single exception, x=0 is used in (N+x). The exception is N=0: then x=1 is used. This requires that the variable GamDistAdd must be set to zero (🡪 Options – Presettings). Under this prerequisite, x=GamDistAdd=0, UncertRadio internally adds 1 to N only if N=0 (this means “0+1”). This means that for N>0 a Gamma distribution is applied for the associated count rate R with a prior(R) ~ 1/R. In contrast, for N=0 a uniform prior(R) is assumed.**

**If, however, x=GamDistAdd has been set a value > 0, which may be true for already existing UR projects, it will always be added to N, also for N>0.**

It is assumed that the (*N*+x) rule may be used only with measurement procedures not requiring a linear least-squares method – and only in the case of very low counting numbers. **For procedures using the least squares method** applied to low counts the fitting result would be questionable; in that case where the Poisson distribution has to be used, the least squares method would be biased and e.g. the Poisson-Maximum-Likelihood-Estimation would have to be applied.

With the following example, based on Gamma-Dist\_EN.txp, it shall be demonstrated that applying the MC simulation may result in asymmetric distributions deviating substantially from the normal distribution.

The equations may be as follows:

A = phi \* Rn

Rn = Rg - R0

Rg = ng / t

R0 = n0 / t0

The (N+1) rule is selected for the quantities ng and n0.

The starting case may be given by: Phi = 1, urel(Phi) = 0,1, t =100 s and t0 = 500 s as well as ng = 8 counts and n0 = 6 counts, i.e. Rg = (8+1)/100=0,09 s-1 and R0 = (6+1)/500=0,014 s-1. The MC-Simulation leads to the following triple graph, showing a slight asymmetry of the simulated (green) distributions:

|  |  |
| --- | --- |
| MCplotfile_a_EN.png | Starting case |

For the next case, with t0 = 500 s, but n0 = 0 counts, especially the distribution of the decision threshold (following triple graph) shows an even more pronounced asymmetry. This asymmetry is the reason that the MC value of the decision threshold is about the double of that of the analytical procedure (blue curve).

|  |  |
| --- | --- |
| MCplotfile-b_EN.png | n0 set to null |

As a third case, compared to the starting case, only the background counting duration t0 is increased to a value of t0=100 x tm=10000 s (a case seldom encountered in practice), which makes the background counting rate R0 very small, especially for the decision threshold a very asymmetric distribution is obtained from which only a very small part is within the negative region. This shape arising from the Gamma distribution is quite different from what one is usually dealing with, e.g., shown under [**MC-Details**](#URH_MCDETAIL_EN).

|  |  |
| --- | --- |
| MCplotfile-c_EN.png | t0 enlarged to t0=100 t |

The reason is given by the fact that with such a small background counting rate the distribution of the decision threshold is mainly affected by the gross counting rate for which the Gamma distribution for very low counting numbers is very asymmetric and always has positive possible values.

## 6.8 Confidence ellipses

Confidence ellipses are invoked from the main menu item “Options – Calculate confidence ellipse”.

**Construction of the ellipse**

The construction of a confidence ellipse of a pair of output quantities is outlined following the GUM Supplement 2 as follows.

At first the covariance matrix **Uy** of two output quantities is determined. For the latter, designated here as *y*1 und *y*2, their covariance matrix **Uy** consists of the diagonal elements and and of the identical non-diagonal elements , with their correlation coefficient .

For the lower triangular matrix **L** of a Cholesky decomposition of **Uy**, indicated by **Uy**=**L** **L**T, the eigenvalues **d** are calculated by the Jacobi method.

The length values **a** of the semi-axes of the ellipse and the angle *θ* between the axes of the ellipse and the axes of the coordinate system are derived from the equations

, *j* =1,2

where is the (1-γ) quantile of the Chi-square distribution with 2 degrees of freedom.

**Graphical realization**

The following figure shows such a confidence ellipse in the left graph. This does not yet correspond with our knowledge of an ellipse, because their principal axes are not vertical. This behavior originates in different scaling units of the two coordinate axes; e.g. 5 scale units show quite different lengths.

|  |  |
| --- | --- |
|  |  |
| both axes have different scales | *re-scale*: both axes have the same scale (5 scale units have the same length on both axes) |

This disadvantage can be removed by introducing the same scale for both axes. This can be achieved by the following re-scaling (in the GUM Supplement 2 it was prevented to use different scaling):

: ,

or

: .

For a graphical presentation the points of the ellipse curve are at first calculated by assuming that the origin of the ellipse is identical with the origin of the coordinate system and that the angle between the axes of the ellipse and of the coordinate system is zero. Applying then a coordinate transformation, consisting of the two operations of a translation and a rotation, are then moved to the final curve of the ellipse which is plotted then.

The right-hand graph of the Figure shown above displays the re-scaled ellipse; their semi-axes are now perpendicular. Within both graphs, the intervals

are indicated as dotted lines.

**Literature:**

JCGM 102:2011 (GUM Supplement 2, 2011)

Brandt, S., 1999, Kapitel 5.10 und A.12

Press et al., 1992, chapter 11.1

M. A. Thompson, 2015: *Gaussian Statistics Lecture*

W. E. Hoover, 1984

## Using data sets for mean and variance

### Mathematical background

Formulae for a mean value and its associated variance presented here are derived by Bayes statistics. Their derivation was described by Weise et al. (2013) in their section 5.8 and their appendix C. Two cases a) and b) are considered (see also Table 2 in [chapter 6.12.1](#URH_Definition_Meantypes_EN)):

**Unknown random influences:**

1. *Mean type 1*. For any input quantity *,* **which does not represent a number of counts**, the variance of *m* individual values is derived from the experimental variation:

(1)

Hierein are:

, and (2)

1. *Mean type 2*. An input quantity *n* represents **a number of counts** and is influenced by an additional variation, e.g., due to repeated sampling and/or chemical analysis, which enlarges the Poisson-derived variance. A normal distribution with parameters and is assumed for this influence. The variance of the mean is then given by:

(3)

and are calculated analogue to und . The variance component

(4)

is considered as the best estimate of the parameter  of the involved normal distribution. The first term in the bracket of Eq. (2), , represents the Poisson-related part of the variance.

Applying these formulae leads to surprising result that a variance can be calculated only if there are more than three individual values.

*Mean type 3*. For comparison purposes the **classical** formula for the standard uncertainty of the mean can be applied

, (5)

if the type of mean “classical“ is selected.

**Known random influences:**

If the fraction of (4) within (3) is small, a parameter can be defined as:

by which Eq. Gl. (3) turns into:

(6)

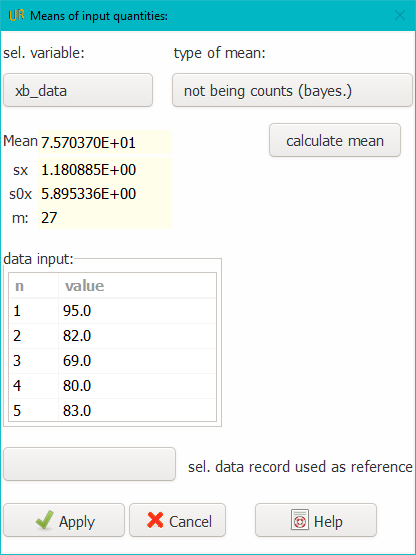
By solving Eq. (4) for , an equation is obtained, by which can be determined from the data set of measurements of a reference sample :

(7)

The parameter value should be less than about 0.2.

### Applying means in UncertRadio

If in the symbol list under the TAB “Equations“ a symbol type is changed into „m“, the program assumes that value and uncertainty of this quantity are to be derived from a data set. The following input dialog allows the input of the data set, it is invoked by the icon  from the toolbar (it requires first selecting the row of this “m” variable in the TAB “Values, uncertainties”):



The id values for the data sets are already known here. In the dialog shown, the id ref\_data (belonging to the input quantity ref) is selected for data input. Besides, the type of mean and variance can be selected from equations (1) and (3). For the extreme case that there are not more than only 3 single values, or the data shall be evaluated in a classical sense, the variance according to Eq. (5) can be chosen as third option. The latter can also be used for more than 3 single values. In the dialog shown, the standard deviations sx and s0x correspond to equations (1) und (2) in 6.9.1.

The combobox indicated in the dialog by the label “sel. data record used as reference“ allows to select one of the mean datasets, which is intended to be used as a reference in the case of “known random influences”. An example project is ISO-Example-2b\_V2\_EN.txp. If no reference data set is selected, the evaluation follows that of the option “unknown random influences“. The details for these options are outlined in section 6.12.

Values of mean and uncertainty of such a data set are transferred by the program to the uncertainty table under the TAB “Values, uncertainties“ by the button “Calculating uncertainties”.

The individual values of this quantity with a name symbol are saved in the project file (\*.txp) as a single line record identified by the associated identification (symbol\_data).

For **organizing the data input** it is recommended to begin with data input into the TAB „Values, Uncertainties“. For mean variables characterized by „m“ as type the „t distribution“ is to be selected as distribution type which enables a correct statistical treatment of the mean within the mean dialog. Then, the mean dialog can be opened in which the desired mean variable is selected; after input of associated singe values the type of mean is selected which then can be calculated. After leaving the dialog the calculation of uncertainties needs to be updated/repeated.

The input of single values in this dialog was modified such, that after input of a value the next cell is already opened for input. It happens that the activated cell appears to be moved a bit away from the grid cell, however, the value entered (finalized with Enter or cursor-down) is transferred into the original grid cell. The input of values is then finalized with typing Enter into the activated cell, which must be empty for this purpose.

## 6.10 Measuring a short-lived radionuclide with comparably long counting duration

### 6.10.1 Basic principles

If the product becomes significantly larger than 0.1, or even , when measuring the activity of a short-lived radionuclide, the Poisson distribution of the gross counts is only an approximation. A feature in this case is that the gross counts distribution is a superposition of a binomial (sample contribution) and a Poisson distribution (background). A characteristic of binomially distributed sample contribution counts is that the variance of the gross counts is smaller than the gross counts itself, i.e., it is smaller than the variance of Poisson distributed gross counts. The binomial distribution for detected sample counts in this context has been applied in the literature (see e.g., Mathews et al, 1979; Spyrou et al., 1981; Salma and Zemplén-Papp, 1992; Gilmore, G., 2008; Semkow, 2007).

Let be the number of atoms existing at the begin of the measurement. The product of the probability for the decay of an atom during the duration and the probability of detecting this decay, constitutes one parameter of the binomial distribution; is the other. relates to the activity by .

The literature mainly restricted the consideration to the binomial distribution of the sample counts contribution. However, the distribution of the gross counts (including background also) is also required. It can be found by folding two discrete distributions, of the binomial and the Poisson variables X und Y:

|  |  |  |
| --- | --- | --- |
| X: |  |  |
| Y: |  |  |
| Z = X + Y: |  |  |

With , and the gross counts , it follows:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

**Note:**

Strictly speaking, the form of the distribution as just defined is valid only for integer values of the binomial parameter . For non-integer values the binomial distribution part is not normalized to one. This problem can be avoided, if Eq. (1) is replaced by a special numerical function, which is expressed by a hypergeometric distribution using the so-called **Kummer confluent hypergeometric function.** This form of distribution allows to apply also non-integer values of . Its computation is also faster than applying Eq. (1), especially for larger values of . This version of the distribution is actually applied in UncertRadio.

For a value of the gross count number to be generated by **Monte Carlo simulation** one binomial distributed value (sample contribution) and one Poisson-distributed value (background contribution) are generated. Both values are added to obtain the gross count number value. However, this follows only that procedure given by Eq. (1) (discrete values), because the binomial-distribution random number generator can only produce integer values, while the Poisson random numbers (background contribution) are continuous. For a small number of gross counts, the shape of the distribution is therefore a series of overlapping peaks, one for each binomial integer value.

### 6.10.2 Aspects of uncertainties and evaluation

The expectation values and the variance of the number of gross counts are calculated by moments of the probability density:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

The corresponding values of the binomial distribution are

und (4)

(5)

which allow to replace in equations (2) and (3) the product by .

By inserting , which can be interpreted as the number of net counts, into Eq. (3), it follows:

(6)

Two quantities being important for uncertainty propagation are the gross and the net count rates. They must be based on directly measured quantities:

**Gross count rate**:

(7)

(8)

**Net count rate:**

One finds for :

**Relation between activity and**

For the activity existing at a number of counts are detected during the counting duration , originating from the radionuclide decay. It is obtained as:

(9)

It follows then from :

or (10)

An equation for follows from the equation for the count rate ( is a correction for the decay during the counting duration)

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

**Activity concentration :**

The relation is used for deriving the activity concentration :

(13)

Based on this equation, the uncertainty is calculated as usual in UncertRadio.

### 6.10.3 An example case

After Ac-228 (half-live (6,15 0,03) h) is radiochemically separated, it is measured during 8 h. For this measurement setup the product is 0.9017. As this is significantly above 0.1, nearly 1, the Ac-228 contribution to the measured gross count rate is considered to follow the binomial distribution. The number of gross counts therefore follow the sum of binomial- and Poisson-distributed contributions.

Symbols and values of input quantities:

*(taken from the UR2 project Ac228\_binomial\_EN.txp)*

|  |  |
| --- | --- |
|  | Number of atoms existing at the begin of measurement ( ) |
|  | : parameter of the binomial distribution:  ; () (1)  0,23764104; 0,004982491  Note: If the measurement of duration does not start at 0, but at , the parameter is extended to: |
|  | detection probability: 0,4 0,0083; |
|  | decay constant of Ac-228, half-live t 6,15 h 0,03 h;  0,1127069 h-1; 5,497896E-04 h-1 |
|  | duration of the Ac-228 measurement (8 h), being not small compared to the half-live |
|  | background count rate, measured with the duration = 20 h: 50 Imp./20 h = 2.50 h-1; |
|  | number of gross counts: 50 counts within 8 h; |
|  | factor converting the activity (Bq) into an activity concentration |

Results obtained by these data:

counts

counts

counts

Bq

further results:

w = 3.79418826 u(w)= 7.90495202E-02 (w0=1)

a = 14.2282066 u(a)= 3.39566064

Sum(Product(Bi x Po)):

mean(BiPo)= 50.0000000 var(BiPo)= 42.8707695

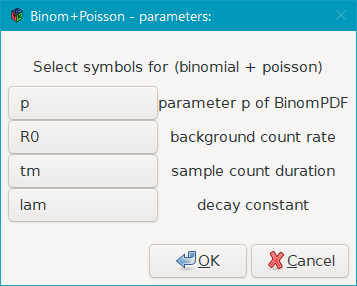
expected Var: Ng\*(1-p) + N0/t0\*tm\*p= 42.8707695

**Note**: The variance of the gross count number (42.871) is smaller than the gross count number (50.0), i.e., smaller compared to a pure Poisson distributed gross count number. By decreasing the detection probability by a factor of 10 results in a 10-fold smaller value of , i.e., . Under this assumption the binomial distribution can be approximated by a Poisson distribution. Then, approximately the relation “variance of gross counts = gross counts“ would be to be expected. This is confirmed by Eq. (6), by which the variance value results in , which is already close to the value of 50 to be expected for the “Poisson plus Poisson“ case.

### 6.10.4 Implementation in UncertRadio

The first step for invoking the specaial procedure for short half-lives is to select under the TAB “Values, Uncertainties“ the distribution type “**Binom+Poiss**“ for the gross counts number symbol.

Thereafter, four further parameters are to be selected: . This can be achieved by invoking a dialog via the menu **Set binomial/poisson case:**



This dialog may also be invoked by the program itself while establishing such an UR project.

The symbol numbers of the four parameters *p*, *R0*, *tm*, *lambda*, are stored in the txp file, e.g., as “BinPoi=8 10 12 9“.

## 6.11 Special distributions and their properties

The following probability distributions are implemented in UncertRadio:

|  |  |  |
| --- | --- | --- |
| distribution | code | notes |
| Normal distribution | Normal |  |
| Rectangular distribution | Rectangle |  |
| Triangular distribution | Triangle |  |
| (N+x) Rule | (N+x) Rule | see section 6.7 |
| Lognormal distribution | LogNormal |  |
| Gamma distribution | GammaDist |  |
| Binomial+Poisson distribution | Binom+Poiss | see section 6.10.1 |
| Beta distribution, 2 parameters | Beta2Dist |  |
| T distribution | T-Distrib |  |
| Beta distribution, 4 parameters | Beta4Dist |  |
| Erlang distribution of the counting duration for preset count numbers | Npreset | is a Gamma distribution for integer numbers of counts |

In addition to better known distributions, like e.g. normal, rectangular or triangular distributions, special distributions can be applied,

* the gamma distribution,
* the beta distribution and
* the *t*-distribution.

the properties of which are described in the following. Their application requires two or three specific parameters.

In the following, the probability density functions (pdf) and the relation of their parameters to measured data are shortly introduced.

The icon  allow to invoke a dialog for showing the distribution-related parameters of an input quantity. This requires that the row within the table “Values, uncertainties” needs to be highlighted.

### 6.11.1 The gamma distribution

Probability density as a function of , with parameters and :

is the gamma function, with

Mean and variance of the probability density are defined by:

;

If mean and variance of measured values are attributed to them, the equations then allow to derive the values of the two parameters:

The gamma distribution can e.g. be used for counting rates or for a detection probability with a larger relative uncertainty.

### 6.11.2 The beta distribution

Probability density as a function of , with parameters and :

is the beta function.

Mean and variance of the probability density are defined by:

;

If mean and variance of measured values are attributed to them, the equations then allow to derive the values of the two parameters:

or (according to NIST):

In contrast to the gamma distribution, the beta distribution is defined within the restricted range . It is thus well suited for a detection probability which normally hast he same support.

### 6.11.3 The *t*-distribution

For convenience, not the standard Student-*t*-distribution is applied, but the “non-standard“ *t*-distribution, also called the “Scaled-and-shifted” *t*-distribution. Its parameters are the number of degrees of freedom and the two parameters (“shift“) and (“scaling“).

The Probability density function:

Usually, this is written as .

Mean and variance of the probability density are defined by:

;

;

For deriving values attributed to the parameters, assume that a series of repeated measurements of normal-distributed values is given, where and are considered as unknown. This leads to the probability density of the input quantity given by the -distribution of the form

with following parameter values ():

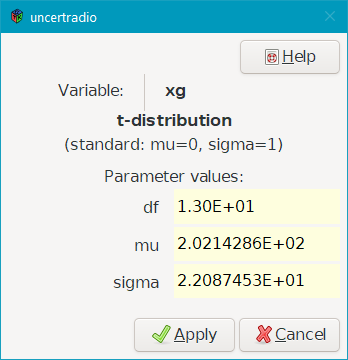
|  |  |
| --- | --- |
|  |  |

This leads to the following values of the expectation values given above:

;

In this case, and are considered as the input values of the *t*-distribution as obtained by measurements, while the factor follows from a property of the *t*-distribution.

As the *t*-distribution refers to a series of measurements, the associated input variable has to be declared as a mean variable in UncertRadio ([see Applying means](#URH_datasets_means_EN)). This guarantees that the parameter values , or , and are known within the program; they may be displayed by the following dialog invoked by the toolbar icon  :



Random values of this distribution density are sampled from , where the are random values of the standard-*t*-distribution, produced by a random generator. The factor must not be part of the formula for ; it implicitly results from applying the -values.

### 6.11.4 Generating random numbers

For gamma-distributed random numbers, the generator by Marsaglia and Wang (2000) and another generator taken from Alan Miller’s repository of Fortran-90 routines ( <https://jblevins.org/mirror/amiller/> ) are applied. For the two other special distributions, also routines from Allan Miller’s repository are used.

## 6.12 Gross quantity: Variance interpolation for a mean

### 6.12.1 Definitions

According to chapter 6.9.1 the following definitions of means and associated variances are applied:

**Table 1:**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

For those input quantities, to which mean values are attributed, the *t*-distribution is taken as type of distribution. The possible values, which can be attributed to the three parameters (number of degrees of freedom , mean value , standard uncertainty (scaling)) of the *t*-distribution, are given in the following table for the example of the gross quantity (subscript g); the table for the background quantity (subscript b) would look similarly.

**Table 2:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Method A**  **“not being counts“** | **Method B**  **“counts, with influence“** | **Method C**  **“classical“** |
|  | ***t*-distribution** | **distribution unclear** | **normal distribution** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

( and are estimated in the correspondent manner like and .)

**Notes:**

The distribution type of the type of mean “counts, with influence“ is a superposition of a shifted t-distribution (mean ) and a normal distribution (mean 0); see section 6.12.2. The case in the third column is program-internally treated as “normal distributed“, even if in UR the *t-*distribution has been chosen as distribution type.

The variance of the sum of a *t-*distributed and a normal distributed quantity is given by the sum of their variances only if more than 5 individual values of the *t-*distributed quantity are used.

If the background-related quantity is not treated as a mean, this meansis applied.

### 6.12.2 Principle of the MC simulation

The theoretical treatment of **Method B** from Table 2 was described by Weise et al. (2013), in its Appendix C, especially section C.2 It is shown there, how the expression for the variance of the mean of the numbers of counts

(1)

was derived. The first term therein, is interpreted as a counting uncertainty contribution. The second term is a *t*-distributed contribution of additional random influences to the variance.

For a normal distribution, the variable

follows a Student *t*-distribution with degrees of freedom, expectation value of zero and variance ; is the expectation value of . Solving this equation for leads to the equation

(2)

This is taken as a recipe for generating t-distributed random numbers. With standard-*t*-distributed random numbers , the MC values for simulating the distribution of according to Eq. (1) are derived as follows.

With

(3)

random values with mean and variance are obtained; in a second step normal-distributed random values are added to this, where are standard-normal distributed random values:

(4)

This last step contributes to broadening the distribution.

For the less complicated case of **Method A,** only equation (2; is replaced by ) is applied for generating random values.

**Notes**:

By using *t*-distributed values the multiplicative factor is generated automatically; therefore, this factor must not be supplied in equations (2) and (3).

In the TAB “Values, uncertainties“ in UncertRadio those uncertainties are displayed, which correspond to the row for in table 2. Before generating MC values for an assumed value according to Eqs (2) or (3), the value is calculated from the associated by reversing equations (2) or (3).

From the uncertainty one calculates:

Eq. (2):

Eq. (3):

**Special feature of the MC simulation of decision threshold and detection limit:**

In these cases, the factor of for the gross count rate is already contained in the expression of its uncertainty varied according to Eq. (11, see below). As already indicated in the notes above, this factor is implied by generating random values : it is identical with the standard uncertainty of the standard distribution. To prevent from applying this factor twice, is simply replaced by in the equations (2) through (4).

### 6.12.3 Procedures with unknown random influences

It is assumed that repeated measurements underly unknown random influences, which are not small and lead to increased fluctuations. This requires running some measurement series for estimating the gross and background count rate (or gross and background quantities).

#### Using the gross scout rate for interpolation

If the **value of a gross count rate Rg or a gross quantity xg** is estimated by a **mean of a measurement series**, its uncertainty can no longer be estimated by, e.g., u(Rg)=sqrt(Rg/t). Instead, this requires an i**nterpolation between two known values of the variance**. According to ISO 11929, this is solved for an assumed value of the output quantity by interpolating between the variance of the primary result and the variance :

(5)

In UncertRadio, however, such an interpolation refers to corresponding two variance values of the gross quantity . This case can be deduced from the one in Eq. (5). A **measurement model with quantities (gross, background, interference)** is assumed, in which both, and are treated as mean values:

(6)

This means

(7)

Equating the right-hand sides of (1) und (3) yields

(8)

With setting

, (9)

It follows:

(10)

Now, with an expression for :

(11)

the expression for the variance becomes:

Setting now :

(12)

For the program-internal application, und are also replaced:

For the last round bracket in (12) one obtains:

(13)

which by inserting it into in (12) yields:

(14)

In principle, equations (12) or (14) represent that equation or formula, which would have to be entered by the user into the “green cell“ in the table “values, uncertainties“ in UncertRadio. This would also imply to add several auxiliary quantities to the symbol list in UncertRadio. However, the already existing tool for treating mean values according to chapter 6.9 (see also section 6.12.), offers the opportunity to gather these auxiliary quantity values internally.

In equation (14), assumed values of the variable are set by the program within the iterations for calculating the decision threshold and the detection limit. The fixed values are taken from the two tables in 6.12., or from the program-internal data arrays associated with the treatment of means. These also fixed values for are read from the UR table „Values, uncertainties“.

Finally, it is no longer necessary to enter a formula into the „green cell“ for the standard deviation of the gross quantity, if the value of this quantity is given by a mean. This requires only to define the *t*-distribution type for the quantity symbol .

**Example projects:** ISO-Example-2a\_EN.txp (with the old UR-treatment)

ISO-Example-2a\_V2\_EN.txp (with the new UR-treatment)

**Equivalence of the linear interpolation alternatives**

The interpolation of output quantity variances to be applied according to ISO 11929 shall be linear as in Eq. (1). As in this section the interpolation instead refers to gross count rate variances, it needs to be tested, whether the interpolated values according to these two interpolation variants would agree. This has been tested with small R program, separately for procedures A and B.

#### Application to Procedure A („not being counts“)

If the variances und are taken as products and according to 6.12.1, for the model case the result for the interpolated variance of the gross quantity is:

(15)

**Testing the variance interpolation:**

(Program Var\_intpol\_Ex13.R, for example 13 of ISO 11929-4)

The following formulae were applied for which, after inserting it into Eq. (7), allows the comparison with variance values calculated according to Eq. (5) ( has been set zero):

var\_Rg\_tilde\_a = fg\*(sb^2) + uxint^2 +

q\_tilde \* (fg\*(sg^2-sb^2)- uxint^2 + xn^2\*(uw/w)^2\*(1- q\_tilde) )

var\_Rg\_tilde\_b = ((fg+fb)\*sb^2 + 2\*uxint^2)\*(1. - q\_tilde) +

q\_tilde\*( (uym/w)^2) - fb\*sb^2 - uxint^2 - q\_tilde^2\*(ym/w)^2\*(uw/w)^2

q\_tilde = (xg\_tilde – xb - xint) / (xg – xb - xint)

With using the following values:

sg= 71.71839 sb= 5.895336 fg= 0.03857143 fb= 0.04012346

xgtilde= 75.704 q\_tilde= 2.542306e-06

xg= 192.25 uxg= 14.08521 xb= 75.7037 uxb= 1.180885

ym= 116.5463 uym= 37.71288 uy0= 1.653796

and Eq. (6) for calculation of xg\_tilde from y\_tilde, for 11 values of ytilde, between 0 and 116.54, the following variances were derived:

y\_tilde xg\_tilde var\_xg\_tilde\_a var\_xg\_tilde\_b vary\_tilde\_lin varytilde2

[1,] 0.00000 75.70370 1.340549 1.340549 2.73504 2.73504

[2,] 11.65463 87.35833 131.068433 131.068433 144.68766 144.68766

[3,] 23.30926 99.01296 236.346847 236.346847 286.64028 286.64028

[4,] 34.96389 110.66759 317.175789 317.175789 428.59290 428.59290

[5,] 46.61852 122.32222 373.555262 373.555262 570.54552 570.54552

[6,] 58.27315 133.97685 405.485263 405.485263 712.49814 712.49814

[7,] 69.92778 145.63148 412.965795 412.965795 854.45075 854.45075

[8,] 81.58241 157.28611 395.996855 395.996855 996.40337 996.40337

[9,] 93.23704 168.94074 354.578446 354.578446 1138.35599 1138.35599

[10,] 104.89167 180.59537 288.710565 288.710565 1280.30861 1280.30861

[11,] 116.54630 192.25000 198.393214 198.393214 1422.26123 1422.26123

There is no difference observed between the two compared variance values of the gross quantity (columns 3, 4). The same observation applies to the output quantity variance (columns 6, 6) calculated according to equations (5) and (7).

This verifies the equivalence of the two compared interpolation methods.

#### Application to Procedure B („counts, with influence“)

For the model , the variances and , calculated according to Eq. (1), are given by:

(16)

with

;

**Testing the variance interpolation:**

(Program Var\_intpol\_Ex14.R, for example 14 of ISO 11929-4)

The following variants of equations for were applied for subsequent comparison with values from Eq. (5) ( has been set zero):

var\_Rg\_tilde\_a = (un0\_mean/t0)^2 + uxint^2 +

q\_tilde \* ((ung\_mean/tg)^2-(un0\_mean/t0)^2 - uxint^2 + Rn^2\*(uw/w)^2\*(1- q\_tilde) )

var\_Rg\_tilde\_b = ((un0\_mean/t0)^2 + uxint^2) \* (1. - 2.\*q\_tilde) +

q\_tilde\*( (uym/w)^2 - q\_tilde\*(ym/w)^2\*(uw/w)^2 )

var\_Rg\_green = urbt^2 + (uRg^2 - urbt^2)\*(Rg\_tilde-rbt)/(Rg-rbt) +

(uw/w)^2\*(Rg\_tilde-rbt)\*(Rg-Rg\_tilde)

with

q\_tilde = (Rg\_tilde - R0 - xint) / (Rg - R0 - xint)

Rg\_tilde = ytilde/w + un0\_mean/t0 + xint

rbt = R0 + xint; urbt = sqrt(uR0^2 + xint^2)

Note that „var\_Rg\_green“ denotes that formula that earlier had been manually inserted into the “green cell” within UncertRadio.

With using the following values:

R0= 0.02723333 u(R0)= 0.002929202 Rg= 0.06798667 u(Rg)= 0.006185528

Rn= 0.04075333 w= 34.39972 uw= 2.786688 ym= 1.401903 uym= 0.261393

uy0= 0.1425014

11 values of ytilde, from 0 to 1.402, were calculated:

y\_tilde var\_Rg\_tilde\_a var\_Rg\_tilde\_b var\_Rg\_tilde\_green

[1,] 0.0000000 8.580222e-06 8.580222e-06 8.580222e-06

[2,] 0.1401903 1.252920e-05 1.252920e-05 1.252920e-05

[3,] 0.2803807 1.626019e-05 1.626019e-05 1.626019e-05

[4,] 0.4205710 1.977321e-05 1.977321e-05 1.977321e-05

[5,] 0.5607614 2.306823e-05 2.306823e-05 2.306823e-05

[6,] 0.7009517 2.614528e-05 2.614528e-05 2.614528e-05

[7,] 0.8411421 2.900434e-05 2.900434e-05 2.900434e-05

[8,] 0.9813324 3.164542e-05 3.164542e-05 3.164542e-05

[9,] 1.1215228 3.406851e-05 3.406851e-05 3.406851e-05

[10,] 1.2617131 3.627363e-05 3.627363e-05 3.627363e-05

[11,] 1.4019035 3.826076e-05 3.826076e-05 3.826076e-05

There is no difference observed between the three compared variance values of the gross count rate.

In the following table, for every value of ytilde, the following calculated values are shown:

var\_y\_tilde\_lin from Eq. (1);

Rg\_tilde (from reversing Eq. (2),

var\_Rg\_tilde (the above mentioned var\_rg\_tilde\_b) ,

varytilde2 (after inserting var\_Rg\_tilde into Eq. (3),

ratio (the ratio varytilde2 / vary\_tilde\_lin)

y\_tilde vary\_tilde\_lin Rg\_tilde var\_Rg\_tilde varytilde2 ratio

[1,] 0.0000000 0.02030666 0.02723333 8.580222e-06 0.02030666 1

[2,] 0.1401903 0.02510862 0.03130867 1.252920e-05 0.02510862 1

[3,] 0.2803807 0.02991058 0.03538400 1.626019e-05 0.02991058 1

[4,] 0.4205710 0.03471254 0.03945933 1.977321e-05 0.03471254 1

[5,] 0.5607614 0.03951451 0.04353467 2.306823e-05 0.03951451 1

[6,] 0.7009517 0.04431647 0.04761000 2.614528e-05 0.04431647 1

[7,] 0.8411421 0.04911843 0.05168533 2.900434e-05 0.04911843 1

[8,] 0.9813324 0.05392039 0.05576067 3.164542e-05 0.05392039 1

[9,] 1.1215228 0.05872235 0.05983600 3.406851e-05 0.05872235 1

[10,] 1.2617131 0.06352432 0.06391133 3.627363e-05 0.06352432 1

[11,] 1.4019035 0.06832628 0.06798667 3.826076e-05 0.06832628 1

This verifies the equivalence of the two compared interpolation methods.

### 6.12.4 Procedures with known random influences

It is assumed that repeated measurements underly unknown random influences, which are small. It is furthermore assumed, that the gross and background and other counts are influenced in the same way, also in the case of different measurements but the same measurement conditions. With a reference analysis, i.e., with a larger number of measurements of a sample (which gets a subscript r), the unknown influence can be quantified by parameter , which is applied also to the other involved measurement quantities like gross and background counts.

The parameter has already been introduced in chapter 6.9.1. It is determined from the reference data set and applied to the uncertainty calculations of the other count numbers (subscript x):

or in the case of gross and background counts as well as for assumed gross counts within the detection limit related iterations:

(1)

(2)

(3)

With applying the tool for means (6.9; see also 6.12.1), the data necessary for calculating , but also those data referring to mean value-related datasets are available within the program. Therefore, the formulae corresponding to the equations (1), (2) and (3) are easily programmed and are part of the program. This means, in contrast to earlier UncertRadio versions, it is no longer necessary for the user to enter uncertainty formulae with the TAB “Values, uncertainties“; the introduction of further auxiliary quantities also is no longer necessary.

In addition, from the datasets supplied to UR, that one of them representing the reference measurements from which has to be derived, has to be identified. This can be done with a combobox field within the dialog shown in chapter 6.9.2.

**Example-projects:** ISO-Example-2b\_EN.txp, Mean-theta\_EN.txp (with the old UR treatment)

ISO-Example-2b\_V2\_EN.txp (with the new UR treatment)

## 6.13 Aggregating activities of several aliquots

In certain cases, the measurement of a sample activity requires to determine the output quantity value from several compartments of the sample, or, regarding a surface contamination, from several measurements covering the entire surface.

This chapter describes how to proceed if several activity measurements, also of different types of measurements, need to be aggregated to obtain a single output quantity value. This value may be calculated as a sum or as an average of the single values. The simple aggregation method as described below also includes the calculation of the associated decision threshold and the detection limit.

If these measurements, however, represent repeated measurements in order to obtain a series of measurement values for a single input quantity, the recommended methods would be those described in section [6.9](#_Using_data_sets) und [6.12](#_6.12_Gross_quantity:).

**Activating the evaluation of several aliquot measurements**

In the text field for equations, the following call is inserted for defining the activity as an aggregation of several values:

*Asum = SumEval(mode, np, A1, A2, …)*

The name of the symbol to the left of the = sign may be freely chosen. The name **SumEval** represents the internal procedure which does the calculations necessary for the aggregation of measured aliquot values. Its arguments are:

*mode* integer number, withe values:

1: calculate the mean value of the individual results,

2: calculate the sum of the individual results,

*np* integer, number of aliquot measurements

*A1, A2, …* the list *np* of symbol names (freely chosen) representing the activity or activity concentration values

Directly following the SumEval call, at first those main equations for calculating the activities Ai are inserted,

Ai = wi \* Rneti , for i=1 to np, one after another

These are followed by the lists of equations defining the calibration factors wi and the net count rates Rneti:

Rneti = Rbi – R0i

It is recommended to use further equations for explaining the count rates by their associated numbers of counts.

A complete example for two aliquot measurements may be defined as follows:

*a = 1/F \* Asum*

*Asum = SumEval(1, 2, A1, A2)*

*A1 = w1 \* Rnet1*

*A2 = w2 \* Rnet2*

*W1 = 1/eps1*

*W2 = 1/eps2*

*Rnet1 = Rb1 – R0*

*Rnet2 = Rb2 – R0*

*Rb1 = Nb1 / tb*

*Rb2 = Nb2 / tb*

*R0 = N0 / t0*

Such factors found in all expressions of wi, may be extracted from the wi, i.e., not included in SumEval, as for instance the factor 1/F (1 / surface area) in the equation above that declaring Asum. This helps preventing covariances between the wi. An input quantity being part of several equations generates covariances between the quantities defined by these equations. This is true for the count rate R0 in the example given above, introducing a covariance between Rnet1 und Rnet2.

Such covariances, however, need not be identified explicitly by the user. They are considered by the uncertainty propagation applied within the *SumEval* procedure in the way, that covariance contributions of the form

induced by the independent input quantities between dependent quantities , are taken into account; refer to section [Kap. 6.1.](#_6.1_Uncertainty_propagation)

**Note**: This procedure does not require further windows dialogs.

**Notes about calculating the decision threshold and the detection limit**

Calculations of the decision threshold and especially the detection limit require to vary the value of the output quantity. Such an iteration step generates a modified value, denoted as . This has to be transformed to new values of the individual values as part of *SumEval.* Two possible ways may be applied, which, based on the sample equations given above, are explained below.

If a mean value is to be calculated from the , a meaningful option would be to set all to the same value . The least-squares method is used as indicated in [section 7.14](#_7.14_Least-squares_calculation) for calculating a weighed mean.

If instead a sum of aliquot values is to be derived, it may be meaningful, to modify the values such that the original ratios between the values are maintained. This may be achieved by applying relative “form“ factors

such that

Then, the modified values – and thereby the gross count rates – are internally calculated from :

The associated uncertainties then are, again referring to the complete example given above:

From the uncertainties of the modified gross count rates, the uncertainty associated with the activity value is calculated. Such pairs are used for the iteration necessary for the detection limit calculation.

**Example project:** sumEval\_sum\_EN.txp, sumEval\_mean\_EN.txp

## 6.14 Application of decay series

### 6.14.1 Basics

The time behaviour of atom numbers of the radionuclides (members) of a decay chain is in principle modelled by a system of first order differential equations. They are called the Bateman equations after H. Bateman’s publication of 1910. Their solution led to atom numbers , which by radioactive decay can be expected after a time duration from the atom number at .

More recent work on improved solutions for such a system applies matrix-based methods, which are applicable in an automated way also for more complex decay chains. The method published by E. Levy in 2019 was chosen for UncertRadio, more specifically the -Matrix introduced in chapter 4 of his publication. It was found that the result of a series expansion of an exponential function of a decay matrix as used before lead to a single matrix, the -Matrix. In environmental radioactivity mainly the activities of radionuclides are of interest instead of their atom numbers. The general relation between activity and atom numbers allows to derive an -Matrix for activities. The rank of the matrix is equal to the number of members of decay chain, denoted as In his chapter 4, Levy derived an algorithm for calculating the elements of the matrix , which is of left-triangular shape. An important feature of is that it holds only for the given time ; for any other time value, always has to be re-computed.

For calculating the elements of the atrix only the decay constants and the values of branching ratios (mit ) of the radionuclides are required. The latter are stored in a separate matrix of rank . After initialisation of this matrix by zero, only those elements are set for which holds. For a decay chain of length and without branching just values exist. The diagonal elements of the matrix contain decay factors (without correcting decay during the measurement duration ) or function values (with correcting decay during the measurement; see chapter 7.8).

For a vector of start activity values given at , the vector of activities after the time duration is calculated by matrix multiplication:

(1)

For the application in UR this is called „forward calculation“.The activity unit used in this equation must be Bq.

When specific activities are to be considered, it must be observed that the associated calibration factors relating count rates to specific activities must not contain such decay corrections which are already part of . If this condition is fulfilled, the matrix equation given above can also be applied for specific activities. The matrix in that case remains the same.

### 6.14.2 Decay corrections

The vector is calculated from by Eq. (1); this step was called „forward calculation”. If the decay chain has only radionuclide decaying to a stable isotope, the inversion of this step („backward calculation“)is often termed „decay correction“: it means the ratio . The activity determined at the time of measurement is back calculated to the date of sampling.

For decay chains of two or more members, the backward calculation becomes more complicated. As an example, consider the decay chain Pb-210/Bi-210/Po-210. The non-trivial problem now consists in deriving the Pb-210 activity at the time of sampling from measurements of the later measurements of the decay products.

For the most elegant solution method of this task, the Eq. (1) is interpreted as a linear least squares problem. The matrix is considered as the design matrix and the vector is treated as the desired solution vector. The vector represents the measurements. A covariance matrix can be established associated with . We assume that this matrix is diagonal.

Under these assumptions the least squares problem can be solved by the following two equations (see CHAGR-ISO-01, section 4.2):

(2)

(3)

The computation of these equations is easily done. The advantage of these equations is that the vector and the associated covariance matrix are practically obtained by the same combined step.

Alternatively, a recursion procedure can be applied. According to the literature [Blobel, Lohrmann, chapter 3.4, Press at al., 1992, section 2.3], the elements of the vector are calculated by a recursive scheme („forward substitution“) from the left triangular matrix (). The scheme is described by the following equations:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Before calculating the uncertainties, the components of in the right-hand side of the equation for need to be substituted by corresponding components of the vector .

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

The calculations according to Eq. (5), which may become more tedious for longer decay chains, are simplified by the LS procedure. The coefficients formally define a matrix , which can be established by Eq. (3):

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

The equations (5) and (6) formulated for a 3-member decay chain are:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Eq. (5) can be re-formulated as follows by tracing back the uncertainties of to the uncertainties of the input quantities :

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

This means, the sensitivity coefficients of the uncertainty propagation are multiplied by factors multipliziert. Dependent on the size of , these factors have the potential to raise the uncertainties of . For the example of the decay chain Pb-210/Bi-210/Po-210, this effect increases by increasing the time interval relative to the half-live of Bi-210.

UncertRadio contains the code for calculating and the equations (1) through (8).

### 6.14.3 Impact on the decision threshold and the detection limit

The ISO 11929-related procedure for deriving the decision threshold and the detection limit needs to be modified in the case of the decay corrections for a decay chain. This is explained for the decay chain Pb-210/Bi-210/Po-210, for which the activity A*1*(0) of the first member of the chain shall be calculated. The common relation A*1*(0) =w\*R*n* between activity and net count rate does no longer apply, because this activity originates from two count rates, those of Bi-210 and Po-210.

The corresponding relation shall be modelled by an equation like A*1*(0)= w*1*\*R*n1* + w*2*\*R*n2.* A modified or assumed activity of the output quantity is obtained by multiplying the primary values by a “modifying factor” :

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

The modified count rate values (=1,2) lead to modified values and uncertainties of the gross count rates, from which the uncertainty is derived, based on uncertainty propagation of Eq. (18). This represents one iteration step in calculating the detection limit. Values for the two „calibration factors“ w*1* and w*2* are not always easily calculated, because they contain also elements of the matrix. They can more generally be derived numerically by the partial derivatives of the output quantity with respect to the net count rates : .

### 6.14.4 Literature

Levy, E.: *Decay chain differential equations: Solutions through matrix analysis*. Computer Physics Communications, 2019, Vol. 234, S. 188-194.

[Blobel, Lohrmann, Kapitel 3.4]

Blobel, V., Lohrmann, E.: *Statistische und numerische Methoden der Datenanalyse*. Teubner Studienbücher Physik. 1. Auflage. Stuttgart: Vieweg+Teubner Verlag, 1998, 358 S. ISBN 978-3-519-03243-4

Kapitel 4.2 in:

Kanisch, G., Aust, M.-O., Bruchertseifer, F., Dalheimer, A., Heckel, A., Hofmann, S., et al.: *Bestimmung der charakteristischen Grenzen bei der Aktivitätsbestimmung radioaktiver Stoffe – Teil 1: Grundlagen.* Version Mai 2022. *CHAGR-ISO-01*

In: Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz, (Hrsg.): Messanleitungen für die Überwachung radioaktiver Stoffe in der Umwelt und externer StrahIung. ISSN 1865-8725. Verfügbar unter: https://www.bmuv.de/WS1517

In Vorbereitung: Kanisch, G., Aust, M.-O., Bruchertseifer, F., Dalheimer, A., Heckel, A., Hofmann, S., et al.: *Zeitverhalten bei mehrgliedrigen Zerfallsreihen.* Version Maixxx 2025. *ZERFALL/MEHRGL*

In: Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz, (Hrsg.): Messanleitungen für die Überwachung radioaktiver Stoffe in der Umwelt und externer StrahIung. ISSN 1865-8725. Verfügbar unter: https://www.bmuv.de/WS1517

### 6.14.5 Implementation in UncertRadio

The implementation requires an additional function for executing the more complex calculations of decay corrections for a decay chain. A function SDECAY is implemented in UncertRadio which is called within the set of user-defined equations for the evaluation of the model. The function call has the form:

Symb0 = SDECAY(fmode, tdiff, tms, avg, Nstart, Ndest, SymbAct1,SymAct2,SymbAct3, …)

SDECAY calculates a value and a standard uncertainty by using the least-squares method outlined above, which are attributed to the variable named Symb0 (which is an activity in Bq).

The meaning of the function parameters are explained in the following table.

|  |  |
| --- | --- |
| Variable | Meaning |
| fmode | Forward calculation (=1) or backward calculation (=0), starting from the respective activities; *integer* |
| tdiff | Time difference; *UR symbol* |
| tms | Measurement duration; *UR symbol* |
| avg | Include (=1) or not (=0) the corrections for decay during the measurement; *integer* |
| Nstart | number of that decay chain member, from which on the decay shall be considered (Nstart >1) of if the full decay chain si to be considered (Nstart=1); *integer* |
| Ndest | number of that decay chain member, the activity of which shall be calculated; *integer* |
| SymbAct1, SymbAct2, … | List of the UR activity symbols of the starting values of the decay chain (fmode=1) or of the end values of the decay chain (fmode=0); one may use more than the 3 activity symbols shown in the call |
| Symb0 | UR symbol of the arrays Messwert and StdUnc, to which the SDECAY function value and standard uncertainty are attributed to |

**Note:** the values of Nstart and Ndest always refer to the full decay chain, even if Nstart > 1 is selected.

Example calls:

Symb0 = SDECAY(fmode, tdiff, tms, avg, Nstart, Ndest, SymbAct1,SymbAct2,SymbAct3)

* cPb210\_t1 = SDECAY(0, t2minust1, tmBi210, 0, 1, 1, cPb210\_t2, cBi210\_t2, cPo210\_t2)

“0“: (fmode): backward calculation;

“t2minusT1“: (diff) time difference t2 – t1;

“tmBi210“: (tms) counting duration (taken from Bi-210);

“0, 1, 1“ mean: “0“: (avg): without correcting decay during the measurement; the first “1“: (Nstart) the decay chain starts from member 1; the second “1“: (Ndest) the decay chain member for which the activity shall be calculated.

“From the activities of the three decay chain members (the last three symbols within the call), given at the time t2, the activity of Pb-210 (the first member: Ndest=1) at t1 (fmode=0) is to be calculated without correcting for decay during measurement; the measurement duration, tmBi210, is not used.”

* cPo210\_t1 = SDECAY(0, t2minust1, tmBi210, 0, 2, 3, cPb210\_t2, cBi210\_t2, cPo210\_t2)

“0“: (fmode): backward calculation;

“t2minusT1“: (diff) time difference t2 – t1;

“tmBi210“: (tms) counting duration (taken from Bi-210);

“0, 2, 3“ mean: “0“: (avg): without correcting decay during the measurement; “2”: (Nstart) the decay chain starts from member 2 (sub-chain Bi-210/Po-210); “3”: (Ndest) the third member of the full decay chain for which the activity shall be calculated.

“The decay chain considered starts from the second member (i.e., the chain Bi-210/Po-210). From the activities of the two decay chain members of three (the last three symbols/parameters within the call), given at the time t2, the activity of Po-210 (Ndest=3) at t1 is to be calculated without correcting for decay during measurement; the measurement duration, tmBi210, is not used.”

If a call of SDECAY is found in one or more equations, a decay chain dialog is invoked:

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

In the upper part of this dialog certain measurement related conditions can be defined. A list box allows the selection of the decay chain from some pre-defined decay chains, in this case, the chain Pb-210/Bi-210/Po-210. The possible condition selections are:

|  |  |
| --- | --- |
| Is a chemical sepa-ration applied? | In the case of combined Sr-89/Sr-90 measurements: yes, the separation of Y-90 from Sr-90 |
| No. (1,2 or 3) of the radionuclide, which is build up since the separation |  |
| Common measurement of the decay chain members on the same detector? | May be yes in the case of beta-emitting radionuclides; depends on the measurement design |
| Number of counting channels (energy windows): | Usually 1; in the case of LSC measurements, there may be used more than one channel |
| Shall decay constants be the input quantities instead of half-lives? | Yes: decay constants used; No: half-lives used |

A few pre-defined decay chains are available in a file **List\_DecaySeries.txt**, which on demand is read by UR:

List of available decay series:

Sr-90-2N : Sr-90 # Y-90 : z12=1

Zr-95-3N : Zr-95 # Nb-95m # Nb-95 : z12=0.0108# z13=0.9892# z23=0.944

Pb-210-3N : Pb-210 # Bi-210 # Po-210 : z12=1# z23=1

The structure of the file is simple:

* every decay chain gets a short name (a string);
* then the nuclide names follow, separated by the character #;
* then the necessary branching ratios zji (with ), which are not zero.

With the button „Transfer selections to Grid“ pre-defined symbol names are transferred to the grid for detection efficiencies (up to three when using more than one counting channel) and chemical yields, which are pre-defined from the radionuclide name. The columns of unused detection efficiencies are left empty.

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

Automatisch generierte Beschreibung

Now, the pre-defined symbols in the table (grid) can be modified. Thereafter, the symbols in the table are merged into the symbol list of the UR project by using the button “implement the dialog data”. For these symbols values and uncertainties have to be inserted in the TAB „Values, uncertainties”.

For a further editing of this dialog at a later time, it can be re-opened from the **Menu Edit – Edit Decay chain**.

### 6.14.6 Generation of decay factor formulas

In the case of radiochemical Sr-89/Sr-90 analyses, formulas for decay factors need to be established. In beta radiation counting both Sr isotopes, Sr-89 and Sr-90, contribute to the same count rate. The decay of Sr-90 is accompanied by an ingrowth of its daughter product Y-90 which also contributes to the count rate. The complete formulas for this application can already be complicated, especially, when corrections for the decay during the measurement are included.

For such an application, UncertRadio supports the user by an additional option for building these formulas as strings which are then transferred into the text field in the dialog for defining the evaluation model of decay curves.

This option can be invoked from the **Menu Edit – Edit decay chain.** This option does not require an SDECAY function call within an UR equation.

For the example mentioned at the begin of this section, the dialogs layout is as follows:

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

Automatisch generierte Beschreibung

With the button “Generate Xi formulas for decay curve fit model“ the corresponding formula strings are generated and transferred in to the associated text field of the model dialog.

**The (coded) result is the following:**

X1 = eSr90A \* fd(tAs+tstart,tmess,lamSr90) + eY90A \* 1/(lamSr90-lamY90)\* ( lamY90\*

( fd(tAs+tstart,tmess,lamY90)-fd(tAs+tstart,tmess,lamSr90)) )

X2 = eSr89A \* fd(tAs+tstart,tmess,lamSr89)

X3 = eSr85A \* fd(tAs+tstart,tmess,lamSr85)

X4 = eSr90B \* fd(tAs+tstart,tmess,lamSr90) + eY90B \* 1/(lamSr90-lamY90)\* ( lamY90\*

( fd(tAs+tstart,tmess,lamY90)-fd(tAs+tstart,tmess,lamSr90)) )

X5 = eSr89B \* fd(tAs+tstart,tmess,lamSr89)

X6 = eSr85B \* fd(tAs+tstart,tmess,lamSr85)

X7 = eSr90C \* fd(tAs+tstart,tmess,lamSr90) + eY90C \* 1/(lamSr90-lamY90)\* ( lamY90\*

( fd(tAs+tstart,tmess,lamY90)-fd(tAs+tstart,tmess,lamSr90)) )

X8 = eSr89C \* fd(tAs+tstart,tmess,lamSr89)

X9 = eSr85C \* fd(tAs+tstart,tmess,lamSr85)

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung

### 6.14.7 Example projects

# 7 Miscellaneous

## 7.1 Implication of changes within the Options menu

Only the **change from TAB “Values, uncertainties” to the TAB “Uncertainty budget”** initiates the calculation of decision threshold, detection limit, result and further statistical parameters (shown later in the TAB “Results“). These calculations depend on the parameter settings in the **menu Options – pre-settings**. If parameters therein have been changed these calculations have to be repeated which is achieved by the above-mentioned TAB change. Take note of the hint occurring then in the right-most status bar field which indicates how to continue then.

## 7.2 Algorithm for numerical calculation of the combined uncertainty:

**Function subprogram UncPropa(p,u):**

! Calculations in double precision

= 0

! Calculations within a DO-loop:

! Begin:

do i=1 to n

! Sensitivity factor = partial derivative with respect to the parameter *pi* :

! Variance contribution of :

end do

! End of the DO-loop:

**Uncertainty budget:** divide all values var(i) by the value *Variance*: these are the relative contributions of the parameter I to the variance!

: This is the value of the **combined standard uncertainty**

## 7.3 Algorithm for iterative numerical calculation of the Detection limit :

Starting value:

i = 0

! Start of iteration loop:

do

i = i + 1

! Let the gross counting rate be the 8th element of the parameter array *p*;

! gross counting rate and its uncertainty, and , respectively, where

! is the background counting rate of the analyte and

! is the blank component of the counting rate due to contributions from further

! sources such as **t**racer **imp**urities

! *z*1 and *z*2 are those by UncertRadio estimated parameters in *y* = *z*1 \* *R*n + *z*2.

; and

EXIT ! Iteration terminated.

end do

! End of iteration loop: the last obtained value is the

! final value of the Detection limit.

## 7.4 Mathematics of the linear LSQ curve fitting with correlated measured values

### 7.4.1 Literature:

ISO 11929:2010, Appendix C.5.4

Klaus Weise a. Wolfgang Wöger, 1999: Meßunsicherheit und Meßdatenauswertung.

Verlag Wiley-VCH Weinheim, in German

S. 200 oben (Section 5.4.2 Lineare Kurvenanpassung)

Roger J. Barlow, 1999: Statistics. A Guide to the Use of Statistical Methods in the Physical   
Sciences. The Manchester Physics Series. John Wiley & Sons Ltd., Chichester, New York.

Section 6.6, pp. 111-113.

For its realisation matrix routines from the Datan-Library are applied (converted to FORTRAN 90):

Datan-Library from:

Siegmund Brandt: Datenanalyse. Mit statistischen Methoden und Computerprogrammen; 4. Auflage. Spektrum, Akademischer Verlag, Heidelberg-Berlin, 1999. In German.

This text book is also available in an English version.

Further references:

T. Hauschild, M. Jentschel, 2001: Comparison of maximum likelihood estimation and chi-square statistics applied to counting experiments. Nucl. Instr. & Meth A 457 (1-2), S 384-401.

S. Pommé & J. Keightley, 2007: Countrate estimation of a Poisson process: unbiased fit versus central moment analysis of time interval spectra. Applied Modeling and Computations in Nuclear Science. In: Semkow, T.M., Pommé, S., Jerome, S.M., Strom, D.J. (Eds.), ACS Symposium Series 945. American Chemical Society, Washington, DC, pp.316–334.2007.ISBN0-8412-3982-7.

T.A. Laurence and B. Chromy, 2009: Efficient Levenberg-Marquardt Minimization of the Maximum Likelihood Estimator for Poisson Deviates. Report LLNL-JRNL-420247, November 13, 2009.

### 7.4.2 Basics

The evaluation model is formulated in the following form (according to Weise & Wöger; *note: the meaning of X and Y is just interchanged compared to Weise & Wöger)*:

The nr-vector characterises the output parameters to be estimated and the n-vector input measurement values to be fitted by the model. The (*n* x *nr*) matrix contains the partial derivatives of the fitting function with respect to the parameters to be estimated which are the functions . *Variances as well as covariances between measured x-values are collected in the non-diagonal uncertainty matrix* .

According to Weise & Wöger the following equations are obtained as solution (see also Appendix C.5.4 of ISO 11929:2010, Eq. (C.32); same notation as here!):

; (5.63)

Here, is the resulting output vector of the fitted parameters and its associated covariance matrix.

For the value of the minimum function one obtains by using :

(5.65)

The covariance matrix can be multiplied with , if the value of the latter is larger than one although it is not allowed from the strict Bayesian point of view and therefore not performed in UncertRadio.

### 7.4.3 Chi-square options

The following literature is recommended:

S. Baker, R.D. Cousins, 1983: Clarifications of the use of Ch-square and likelihood functions. Nucl. Instr. & Meth 221, S 437-442.

T. Hauschild, M. Jentschel, 2001: Comparison of maximum likelihood estimation and chi-square statistics applied to counting experiments. Nucl. Instr. & Meth A 457 (1-2), S 384-401.

The calculations shown under Basics use the “***Neyman Chi-square“* expression,** defined as follows (mnemonic: **WLS**):

,

i.e., the diagonal elements of are the variances of the measured values . This may lead to a bias of ***y*** in the case of very low counting rates. For reducing the bias the so-called “***Pearson Chi-square“*** expression can be used which is defined as (mnemonic: **PLSQ**):

The variance given in the nominator is calculated with using the function to be fitted.

,

where and represent the background or net blank count rate, respectively.

As this function value can only be calculated from the value of ***y*** which is to be fitted, the minimization of the chi-square expression requires an iterative procedure. The iteration can be written in analogy to the Eq. (5.63), with (m) and (m‑1) being two consecutive steps:

;

Before repeating an iteration, the “measured“ variances need to be replaced from the fitting function by re-calculating the diagonal elements of with using the values from the preceding step (the non-diagonal elements are not changed):

**Note**: If the covariances between measured values are equal to zero, i.e. the matrix is diagonal with variances as its elements, this procedure completely corresponds to the better known *weighted linear least-squares fitting method.*

As an alternative to the Pearson Chi-square expression the so-called „**Poisson Maximum Likelihood Estimation**“-procedure (or Poisson MLE) may be used for reducing the bias. The associated Chi-square expression is (mnemonic: **PMLE**):

.

The minimization of this expression requires a non-linear fitting method: in UncertRadio a modified version of the “Levenberg-Marquardt“ fitting routine Lm established and described by H. P. Gavin (2022; The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems).The Matlab code of Lm was converted to Fortran. Hints about the modification of Lm necessary for Poisson MLE can be taken from the report LLNL-JRNL-420247 (2009).

The PMLE method can be selected in the example project vTI-Y90-16748\_BLW\_V2\_EN.txp.

While usually net count rate curve are being fitted, this procedure requires gross counts to be fitted. This small transformation is done within UncertRadio; it has the advantage that gross counts are always non-correlated, which may be different when working with net count rates.

According to Baker & Cousins the application of the **PMLE method preserves the area under the curve to be fitted,** while PLSQ tends to overestimation and WLS to underestimation.

[**Notes on applying the PMLE procedure**](#URH_PMLE_Hinweise_EN)

[**Notes on the basics of the non-linear PMLE procedure**](#URH_PMLE_Grundlagen_EN)

**Statistical tests of Chi-squared options**

The function to be fitted is represented by the sum of two gaussian peaks with equal area and identical width (sigma = 5.1 channels, net peak peak areas of 150 counts each, located at channels 30 and 90) on a constant background of 4 counts per channel. Thus, the fit function has 6 parameters, which are considered as true values: 4.0, 150, 30, 5.1, 150, 90. These are considered as true values, from which a large number of Nsp=400 spectra are simulated, with their „true“ channel counts being replaced by Poisson distributed counts (noisy). Each of these spectra are fitted with the different fitting methods using non-linear as well as linear fitting. The following figure shows such a spectrum, with the “true” fitting function and those obtained by the non-linear methods WLS and PMLE.

|  |
| --- |
| Ein Bild, das Text, Diagramm, Reihe, Screenshot enthält.  Automatisch generierte Beschreibung |

The evaluation of 400 spectra in each case, which are fitted non-linearly by the methods WLS, PLSQ and PMLE, is tabulated in the following. In each case, from 400 fitted values of the 6 parameters, a mean value (pa=), a standard deviation between the 400 values (sd=) and mean value of the individual the standard uncertainties (mean(sds..)), calculated by the fitting routine, are derived. In addition, sums (over the 6 parameters) of absolute deviations (fitted value minus true value) and of differences (sd minus sds) are calculated, which are given in two additional lines below each single table. These values serve as good indicators of the consistency of fitting.

Non-linear fitting:

mean values: for method WLS

i=1 pa= 2.75534E+00 sd(pa(i))= 2.34472E-01 mean(sds(i))= 1.73935E-01

i=2 pa= 1.57094E+02 sd(pa(i))= 1.91809E+01 mean(sds(i))= 1.59540E+01

i=3 pa= 2.99839E+01 sd(pa(i))= 7.59367E-01 mean(sds(i))= 5.81371E-01

i=4 pa= 5.18531E+00 sd(pa(i))= 5.09876E-01 mean(sds(i))= 3.88957E-01

i=5 pa= 1.58079E+02 sd(pa(i))= 1.84934E+01 mean(sds(i))= 1.59980E+01

i=6 pa= 8.99873E+01 sd(pa(i))= 7.84781E-01 mean(sds(i))= 5.79122E-01

sum of absolute deviations from true(pa) : 16.5316238

sum of absolute deviations of two sd types : 6.28735828

mean values: for method PLSQ

i=1 pa= 3.99617E+00 sd(pa(i))= 2.11079E-01 mean(sds(i))= 1.73482E-01

i=2 pa= 1.49851E+02 sd(pa(i))= 1.75574E+01 mean(sds(i))= 1.57870E+01

i=3 pa= 3.00262E+01 sd(pa(i))= 6.38806E-01 mean(sds(i))= 5.91121E-01

i=4 pa= 5.07812E+00 sd(pa(i))= 4.50239E-01 mean(sds(i))= 3.95102E-01

i=5 pa= 1.50346E+02 sd(pa(i))= 1.63419E+01 mean(sds(i))= 1.58062E+01

i=6 pa= 8.99958E+01 sd(pa(i))= 6.39568E-01 mean(sds(i))= 5.90819E-01

sum of absolute deviations from true(pa) : 0.551150203

sum of absolute deviations of two sd types : 2.49521804

mean values: for method PMLE

i=1 pa= 4.00893E+00 sd(pa(i))= 2.10194E-01 mean(sds(i))= 2.07077E-01

i=2 pa= 1.51928E+02 sd(pa(i))= 1.69198E+01 mean(sds(i))= 1.68943E+01

i=3 pa= 2.99583E+01 sd(pa(i))= 6.46945E-01 mean(sds(i))= 6.26552E-01

i=4 pa= 5.08396E+00 sd(pa(i))= 4.22725E-01 mean(sds(i))= 4.34723E-01

i=5 pa= 1.49174E+02 sd(pa(i))= 1.73843E+01 mean(sds(i))= 1.67870E+01

i=6 pa= 8.99848E+01 sd(pa(i))= 6.22015E-01 mean(sds(i))= 6.35672E-01

sum of absolute deviations from true(pa) : 2.83508420

sum of absolute deviations of two sd types : 0.671966136

The best absolute deviation for (Fit value minus true value) is obtained for the PLSQ method; the best consistency of fitted uncertainties (abs. deviation (sd minus sds)) is found for the PMLE method.

That in this analysis the methods PLSQ and PMLE yield better consistency values than the classical fit (WLS), can be explained by the condition of having a quite low background of 4 counts per channel. If this background would be significantly increased, the difference between the fitting methods would be much smaller.

The spectra can also be treated by linear fitting methods, if the width parameter and the two peak position parameters are held fixed at their true values. In this case, only the background parameter and the two peak areas are fitted. The results of such an evaluation are given in the following.

Linear fitting:

mean values: method = WLS

i=1 pa= 2.77091E+00 sd(pa(i))= 2.19480E-01 mean(sds(i))= 1.61749E-01

i=2 pa= 1.53462E+02 sd(pa(i))= 1.78632E+01 mean(sds(i))= 1.50700E+01

i=5 pa= 1.54138E+02 sd(pa(i))= 1.74788E+01 mean(sds(i))= 1.51000E+01

sum of absolute deviations from true(pa) : 8.82905006

sum of absolute deviations of two sd types : 5.22971344

mean values: method = PLSQ

i=1 pa= 4.00339E+00 sd(pa(i))= 1.95047E-01 mean(sds(i))= 1.92895E-01

i=2 pa= 1.49325E+02 sd(pa(i))= 1.63338E+01 mean(sds(i))= 1.58374E+01

i=5 pa= 1.49789E+02 sd(pa(i))= 1.53364E+01 mean(sds(i))= 1.58544E+01

sum of absolute deviations from true(pa) : 0.889460266

sum of absolute deviations of two sd types : 1.01658702

Similarly, as for non-linear fitting, the results for the PLSQ method show a better consistency than for the WLS method.

### 7.4.4 Export of input data to R

With UncertRadio it is possible to export the input data, depending on the chosen fitting procedure (apart from WTLS: **R** does not yet support this!), in a format which as compatible with the corresponding R routine into a text file (URExport-to-R.txt, or similarly). This file / these files can easily be imported by the **statistics package R.** This allows comparing the results between UR and R. This option is invoked by the menu item “options – LSQ export to R” and can be used for the cases of calculating the output quantity and the decision threshold. The data required for this refer to counting rates only, not to the result for the output quantity. For the case of the output quantity (see URExport-to-R.txt) one obtains:

(Note: the covariance shown below is truncated at the right hand)

**Case: output quantity**

Blank count rate= 4.66670009E-08 background rate= 1.88333332E-03

Input data: variance-covariance matrix: (rank= 18 )

2.87780E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.46788E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 1.96152E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 1.64805E-07 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 1.41898E-07 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 1.35870E-07 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 1.43104E-07

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

Arrays y, X1, x2, X3:

y X1 X3 (Eingangsdaten-Matrix)

1 5.65134E-03 8.80220E-01 2.73093E-01

2 4.47079E-03 8.07210E-01 1.10866E-01

3 3.01245E-03 7.40257E-01 4.50075E-02

4 2.10968E-03 6.78857E-01 1.82714E-02

5 1.44995E-03 6.22550E-01 7.41751E-03

6 1.27634E-03 5.70913E-01 3.01124E-03

7 1.48468E-03 5.23559E-01 1.22245E-03

8 9.98564E-04 4.80133E-01 4.96272E-04

9 8.47116E-04 4.40700E-01 2.03089E-04

10 8.24953E-04 4.03787E-01 8.17887E-05

11 1.24162E-03 3.70295E-01 3.32032E-05

12 5.12453E-04 3.39582E-01 1.34793E-05

13 9.63842E-04 3.11415E-01 5.47210E-06

14 3.38842E-04 2.85585E-01 2.22147E-06

15 1.65231E-04 2.61897E-01 9.01838E-07

16 -7.78244E-05 2.40175E-01 3.66113E-07

17 2.69398E-04 2.20253E-01 1.48629E-07

18 1.99953E-04 2.01985E-01 6.03378E-08

Parameter values and std uncertatinties obtained by UR:

1 2.83190E-03 3.55440E-04

3 1.45234E-02 2.01819E-03

Chisqr= 1.23143363

For the import to R the covariance matrix and the input data Are written to separate files:

Output quantity: covmat1.txt and data1.txt

Decision threshold: covmat2.txt and data2.txt

With these files a statistical evaluation by R can be done as follows:

(load package MASS) (R)

One obtains with R for the output quantity:

> covmat <- read.table("covmat1.txt")

> data <- read.table("data1.txt")

> res <- lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

> summary.lm(res)

Call:

lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

Residuals:

Min 1Q Median 3Q Max

-8.076e-04 -4.422e-04 -3.702e-04 -3.134e-05 5.747e-04

Coefficients:

Estimate Std. Error t value Pr(>|t|)

X1 2.832e-03 1.643e-07 17231 <2e-16 \*\*\*

X3 1.452e-02 9.332e-07 15564 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0004624 on 16 degrees of freedom

Multiple R-squared: 0.9625, Adjusted R-squared: 0.9578

F-statistic: 205.1 on 2 and 16 DF, p-value: 3.95e-12

Warning message:

In summary.lm(res) : calling summary.lm(<fake-lm-object>) ...

To be able to compare the uncertainty with that given by UR, the uncertainty from R (Std. error) is divided by the value Residual standard error.

Thus, one obtains from R an uncertainty of the fitted parameter X1:

1.643E-07 / 0.0004624 = 3.5532E-04

Case: Decision threshold

Blank count rate= 4.66670009E-08 background rate= 1.88333332E-03

Input data: variance-covariance matrix: (rank= 18 )

2.29269E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 1.47460E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 1.14249E-07 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 1.00767E-07 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 9.52931E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 9.30711E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 9.21690E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08 2.61574E-08

Arrays y, X1, x2, X3:

y X1 X3

1 3.96624E-03 8.80220E-01 2.73093E-01

2 1.61015E-03 8.07210E-01 1.10866E-01

3 6.53662E-04 7.40257E-01 4.50075E-02

4 2.65363E-04 6.78857E-01 1.82714E-02

5 1.07728E-04 6.22550E-01 7.41751E-03

6 4.37335E-05 5.70913E-01 3.01124E-03

7 1.77542E-05 5.23559E-01 1.22245E-03

8 7.20756E-06 4.80133E-01 4.96272E-04

9 2.94955E-06 4.40700E-01 2.03089E-04

10 1.18786E-06 4.03787E-01 8.17887E-05

11 4.82232E-07 3.70295E-01 3.32032E-05

12 1.95772E-07 3.39582E-01 1.34793E-05

13 7.94798E-08 3.11415E-01 5.47210E-06

14 3.22691E-08 2.85585E-01 2.22147E-06

15 1.31030E-08 2.61897E-01 9.01838E-07

16 5.32199E-09 2.40175E-01 3.66113E-07

17 2.16298E-09 2.20253E-01 1.48629E-07

18 8.80329E-10 2.01985E-01 6.03378E-08

Parameter values and std uncertatinties obtained by UR:

1 1.98944E-11 3.10543E-04

3 1.45234E-02 1.73864E-03

Chisqr= 8.88178420E-16

For the case of the decision threshold one obtains with R

(in this case, only the uncertainty associated with X1 is value of interest; the value of X1 should be close to zero):

> covmat <- read.table("covmat2.txt")

> data <- read.table("data2.txt")

> res <- lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

> summary.lm(res)

Call:

lm.gls(formula = y ~ X1 + X3 - 1, data = data, W = covmat, inverse = TRUE)

Residuals:

Min 1Q Median 3Q Max

-1.370e-09 -5.200e-13 -4.300e-13 3.721e-11 8.881e-10

Coefficients:

Estimate Std. Error t value Pr(>|t|)

X1 2.193e-11 1.334e-13 1.644e+02 <2e-16 \*\*\*

X3 1.452e-02 7.467e-13 1.945e+10 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.295e-10 on 16 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: 1

F-statistic: 5.105e+13 on 2 and 16 DF, p-value: < 2.2e-16

Warning message:

In summary.lm(res) : calling summary.lm(<fake-lm-object>) ...

Thus, one obtains from R an uncertainty of the fitted parameter X1:

1.334E-13 / 4.295E-10 = 3.10594E-04

### 7.4.5 Note on the calculation of Decision threshold and Detection limit:

These special output quantities refer to the component of the result vector and its uncertainty , which is iterated until and fulfil the terminating condition of the iteration. is replaced by one new value determined by the iteration. From this one obtains according to the model , which yields a modified covariance matrix . For this purpose, at first new values are attributed to the **gross counting rates** of the decay curve:

From this the **Uncertainty function (standard uncertainty) of the gross counting rate** results:

**.**

The net counting rates of the modified decay curve then are:

,

from which the diagonal elements of the varied covariance matrix (variances of the net counting rates) result:

,

while the non-diagonal elements are left unchanged.

From the right-hand formula of Eq. (5.63) above the uncertainty is determined. With the pair and the next iteration step can be started; this iteration may be repeated if the convergence criterion is not yet met.

## 7.5 Note on Decision threshold and Detection limit for linear fitting

At first, for the equation used in the Least squares fitting

it must be defined, which of the fitting parameters corresponds to the actually valid output quantity. For the following notes let us assume that the parameter is the one representing the output quantity.

The procedure for calculating the Decision threshold and the Detection limit follows ISO 11929:2010. The parameter is modified by iteration. At first, the parameter in the above equation (i.e. the value of the Y-90 counting rate in the case of a Y-90 decay-curve) is replaced by an iterated value while all other values remain unchanged. From this, new measured net counting rates are calculated as well as its new uncertainties (the uncertainty function). With these new values, i.e. (,,), the least squares analysis calculations are repeated yielding the uncertainty of the iterated parameter value . With this pair of values (,) it is then tested whether the termination condition of the iteration procedure is fulfilled; if not, the next iterated value is determined and the above procedure is repeated in order to find its uncertainty ; and so on.

A more detailed description of these calculations while iterating can be found at the end of the following help topics:

a) [**Mathematics of linear curve fitting with WLS**](#URH_LLSQ_MATHE_EN),

b) [**Mathematics of linear curve fitting with WTLS**](#URH_GLSQ_MATHE_EN).

## 7.6 Notes on linear curve-fitting using general least squares (WTLS)

The so-called “general case” of the method of least squares (WTLS) may be used in UncertRadio in such cases where uncertainties but also covariances are attributed to the values. It is the only method known to the author which allows the inclusion of such uncertainties and covariances.

For its realisation matrix routines from the Datan-Library are applied (converted to FORTRAN 90), the main subroutine being LSQGEN:

Datan-Library from:

Siegmund Brandt, 1999: Datenanalyse. Mit statistischen Methoden und Computerprogrammen; 4. Auflage. Spektrum, Akademischer Verlag, Heidelberg-Berlin. In German.

This text book is also available in an English version.

Although the model continues to be linear with respect to the fitting parameters, WTLS requires an iterative procedure. Therefore, starting values of the fitting parameters are obtained from a preceding call to the simpler LSQ routines.

Unfortunately, the mathematics used in the routine LSQGEN is rather complicated, which is the reason that a description cannot yet be given here. Another routine precedes LSQGEN in UncertRadio which assembles the input data in a special form which is required by LSQGEN. Measured net counting rates and the different values are combined in one common vector y; the same applies to the uncertainties and covariances of the and values, which results in a common input covariance matrix cy. The rank of the vector and of the quadratic matrix cy may easily increase to a significant value. If, for example, a LSC measurement with 3 counting channels is considered and 10 measurements are done, one obtains:

3x10 = 30 values (10 values for each of the channels A, B and C)

3x3x10 = 90 values (3 values associated with each of the values )

Therefore, the rank will have the value 30+90=120. This means, that the input covariance matrix becomes a 120x120 matrix, which makes it easily plausible that the iteratively working routine LSQGEN may indeed become more time-consuming.

**Note on the calculation of Decision threshold and Detection limit:**

To be comparable now with the description of the LLSQ mathematics the result vector of the fitted parameters will here be designated as .

The special output quantities to be considered here refer to the component L of the result vector of the fitted parameters, , and its uncertainty , which is iterated until and fulfil the terminating condition of the iteration. is then replaced by one new value determined by the iteration. Using this one, one can calculate new values of the **gross counting rates** of the decay curve, where L=1 was set without losing generality:

.

From this the **uncertainty function (standard uncertainty) of the gross counting rate** results:

**.**

The net counting rates of the modified decay curve, i.e. the new , then are:

,

from which the diagonal elements of the varied covariance matrix result:

,

while the non-diagonal elements are left unchanged.

With these modified values and uncertainties of the net counting rates the evaluation for the output quantity is repeated yielding the new uncertainty value . Now, the convergence criterion can be tested; if it is not yet fulfilled, the next iteration step is initiated by determining the next iteration value from the pair and .

## 7.7 Within tables: delete rows, working with column blocks

Having already obtained a symbol table form the interpretation the equations, a **subsequent modification of the equations may result in new symbols or other symbols may become redundant**, which leads to changes in the symbol table.

**Additionally occurring new symbols** are indicated in the symbol table by **green colored rows**. For these, the columns *Unit* and *Meaning* may be edited.

Some **symbols may have become redundant**, because they are no longer used; the corresponding symbol rows are shifted to the end of the table and indicated by **yellow color**. Here, the user must decide whether these rows can be deleted; if so, these can be deleted by clicking the toolbar icon  .

At present, **Column blocks** cannot be defined within tables; corresponding blocks of data can therefore not be exported to, e.g., Excel. It is however possible by “copy and paste” to **insert such a column block in an UR table** which has been selected in Excel and copied to the Windows clipboard. This requires the following steps:

* open the upper-left cell of the UR table by double-clicking into it;
* insert the block into this cell using “paste” in the mouse context menu; In this moment, this cell holds the whole block.
* pressing the enter button once extends this block over the area of cells.

## 7.8 Text field for equations

Equations can be written into this text field line-by-line. A special end-of-line character is not necessary; only in the case that **an equation has to be continued in additional lines** each (but not the last) line of this equation must have a “&” character at its end.

The **equations** must be **set up** in a **hierarchical** way.

One **starts with that basic equation** which defines the **output quantity *y***. This may for example read:

*y = w \* Rn - Ai*

Naturally, another symbol can be used for the output quantity. In the following lines for those symbols used only in the right-hand parts of the preceding equations, if they do not already represent a primary input quantity, further equations are defined (**secondary equations**), for example:

*Rn = Rg - R0* net counting rate

w *= 1. / (eps \* eta \* m) \* f1* procedure dependent calibration factor

*f1 = exp(+log(2.) \* t1 / tr)* inverse decay factor

*Ai (=z2)* an interference contribution to be subtracted

Herein, *Ai* represents an interference contribution to the activity, i.e. *Ai* equals the constant *FC* determined internally by the program. The factor *w* corresponds to the other constant, *FL.*

Notes: a) If more than one output quantity were defined for the project, e.g. three, then for each output quantity one basic equation must exist; these then are the first three equations. b) In such cases where interference by another radionuclide exists, *Rn* must be understood as the “**procedure dependent net counting rate**” the equation of which contains an extra term calculated for this interference.

*The simple expression Rn = Rg - R0 may be used also in the case of additional interference contributions. The latter (interference) contribution is taken automatically into account by the internally determined auxiliary quantity FC (*[*see also*](#URH_VERFAHREN_EN)*).*

Because of the hierarchical structure, the **equations are evaluated from bottom to top for obtaining values** for all the quantities. This means that in any equation only such symbols can be used in it belonging to secondary (auxiliary) equations following that equation. The program internally tests whether this condition is fulfilled; if not, the user will get an associated warning.

It is necessary to **use explicitly an equation defining the net counting rate** *Rn.* In this important equation it is allowed for the symbol of the gross counting rate to be multiplied with a factor; in seldom cases, this may be necessary. The value of this factor is identified by the program; it only may play a role for determining Decision threshold and Detection limit.

Note: The gross counting rate symbol must be directly contained in the equation defining the net counting rate, or, another symbol in the latter equation points to a further auxiliary equation in which then contains it.

Example: *Rn = Rn1 - Rblank*; *Rn1 = Rg - R0*.

The **procedure dependent factor *w*** in the above example contains the inverse decay factor *f1* for correcting the radioactive decay of a radionuclide r, having the half-live *tr,* in the time duration *t1* between sampling and the beginning of the measurement. The detection efficiency, chemical yield and sample mass are *eps*, *eta* and *m,* respectively.

The **symbols** occurring in the equations to the left of the equation sign are classified as **“dependent (a)”**, those of the symbols of the right-hand sides and not occurring somewhere left of the equation sign, as **“independent (u)” input quantities**.

It is possible to make full use of secondary equations. By doing this, in the conventional way of uncertainty propagation it happens that **easily overlooked covariances between dependent quantities occur** **when using their uncertainties for propagation**. However, **this cannot happen in UncertRadio**, because it uses only uncertainties from independent quantities.

The **syntax for writing formula symbols** should be the **same as for creating variable names in programming languages**. The program here uses FORTRAN 90 internally. It is not differentiated between lower and upper-case characters. However, it is recommended to the user to make this differentiation for a better readability of the equations. The use o f the underscore (\_) is allowed within symbol names, but not for the first character of a name. A formula symbol must always begin with an alphabetic character.

For numbers occurring in equations as well as in tables the **decimal character** must always be a dot (decimal point). Numbers in equations, e.g. 1. and 2. within the equations for *Fact* and *f1* shown above, are interpreted always as double precision numbers internally by the function parser.

**Internal functions and operators:**

All internal calculations are done with "double precision" arithmetic.

The following intrinsic arithmetic functions can be used, similarly - but not fully identical - as in MS Excel:

sqrt(x) square root function

exp(x) exponential function

log(x), ln(x) natural logarithm

log10(x) common logarithm

A new function fd() with three parameters can be used which calculates a decay factor averaged over the counting duration:

fd(tA,tm,xlam) = exp(-xlam\*tA) \* (1 - exp(-xlam\*tm)) / (xlam\*tm)

It also is:

fd(tA, tm=0, xlam) = exp(-xlam\*tA)

fd(tA=0, tm, xlam) = (1 - exp(-xlam\*tm)) / (xlam\*tm)

This function did not exist in UR1.

Some projects may require applying an uncertainty u(x) of an input quantity value x as an own value. A **function uval(x)** was therefore introduced by extending the function parser. As an example, the relative uncertainty can be introduced as a variable urelw as follows:

urelw = uval(w) / w

The argument of the function uval() must be an existing single symbol taken belonging to the symbol table. An arithmetic expression of more than one variable is not allowed; the latter case, e.g. uval(a+b), would mean to perform an uncertainty propagation for such an expression, what uval() is not made for. If the value of uval(x) shall be treated as a constant value, x must not represent a gross count or gross count rate, because their values and uncertainties vary during calculating the decision threshold and the detection limit.

In addition to conventional operators +, -, \* and /, for the exponentiation one can use \*\*: a\*\*b means a to the power of b, for which writing a^b is also allowed.

**Notes:**

The program already contains a procedure which allows **estimating the net counting rate as a result of weighted multi-linear Least squares fitting applied to a measured decay curve**. This is available for decay curves of Y-90 and may be easily applied e.g. to combined build-up/decay curves measured in a source containing Y-90, Sr-89 and Sr-90.

*This tool is not yet in its final state. Therefore, it is necessary to consider further applications; tips about such examples would be highly acknowledged by the author!*

Further information: [**use of Least squares fit**](#URH_LSQ_EN)

For the **field of gamma spectrometry** there is a procedure available allowing **the activity of a radionuclide with several gamma lines to be estimated as a mean of single line activities.** Two methods for calculating means are offered.

The first method is that of the **weighted mean**, for which so-called “internal” and “external” standard deviations can be calculated. If the values of the two standard deviations are of quite similar size, one can draw the conclusion that the single line activity values are under “statistical control”. This is a well-known procedure; however, it should be noted that the use of the “external” standard deviation is not really Bayes conform.

The second method uses a matrix-based least squares procedure instead of formulae for the weighted mean. It is better suited for including covariances.

This method can only be used, if the gamma lines used for calculating the activity of the radionuclide are not interfered by gamma lines belonging to other radionuclides.

Further information: [**Activity calculation from several gamma lines**](#URH_GSPK1_EN)

## 7.9 Editing the symbol list

Clicking the button “Load symbols from equations” enables the **symbol table for editing** while the **correct syntax of the equations** is being checked with the internal function parser. In the case of an error corresponding warnings are given via Windows Message-box dialogs, which are acknowledged, and one has then look for reason behind that error message.

The symbol table consists of four text columns "**Symbol**", "**Type**", "**Unit**" and "**Meaning**".

The symbols which were extracted automatically are characterised in the column “Type” as dependent (a) or independent (u) quantities, where the independent ones are listed first. **Only the independent symbols (quantities) are the ones the measurement uncertainties of which are considered by the program. This is the important point which prevents at the very beginning that covariances could otherwise be inferred between dependent symbols because they would share common (independent) symbols.**

Within the symbol list, such input quantities for which a mean and its uncertainty are to be derived from a data set, can be marked in the type column with “m“ instead of “a“ (dependent) or “u“ (independent). Furthermore, a quantity, to be derived by an equation, which shall be treated as a parameter without uncertainty, can be defined by sitting its type to “p“.

The columns “Unit” and “Meaning” are to be completed by the user. However, it is recommended to this later, at least after the equations have got their final state, because subsequent changes in the equations may lead to a different ordering of the symbols. This may also be postponed to a later session.

Due to subsequent changes within equations it may happen, that some of the symbols do not occur in the present set of equations. Such symbols are listed at the end of the symbol list, where they can be row-wise removed by the user (see below: Notes on editing tables).

After a mouse click inside a table **mouse wheel scrolling** is possible.

**Very important:**

Most often it is necessary to add manually such symbols at the end of the symbol table which do not explicitly occur in the equations but are needed within those formulae describing standard uncertainties; e.g. counting times tm or t0 are required to calculate the standard uncertainty of gross or background counting rates, sqrt(Rg / tm) or sqrt(R0 / t0), respectively.

**See** [**Notes on editing tables**](#URH_TABTRICKS_EN)

## 7.10 Dialog “Values of decay curve”

The following picture gives an overview of the structure of the dialog.

|  |
| --- |
|  |

Initially, in this dialog **date and time of the Y-90/Sr-90 separation** is entered using the format “DD.MM.YYYY HH:MM:SS”, e.g. “07.09.2006 09:04:00”. It is recommended to use the 4-digit version of the year. The program internally replaces all decimal dots by blanks and then reads the six date/time elements format-free.

One can choose between s and min for the **basis time unit for the counting time**. Internally, if min is chosen by the user, this is converted to the basic unit s.

The value of the net blank counting rate shown (disabled) in this dialog is that of the symbol *Rbl*, which has been entered by the user within the TAB „Values, uncertainties“. It must not contain a detector background contribution.

If the case of selecting **absolute time indication**, complete date and time values are to be inserted in the cells in the table column “Start date (gross)“. The program internally calculates the time differences to the time of the Y-90/Sr-90 separation. **Otherwise,** the time differences are to be inserted into this column directly, only in the unit of seconds.

In the seldom case, that decay corrections are not considered, one may simply insert ongoing numbers into this column.

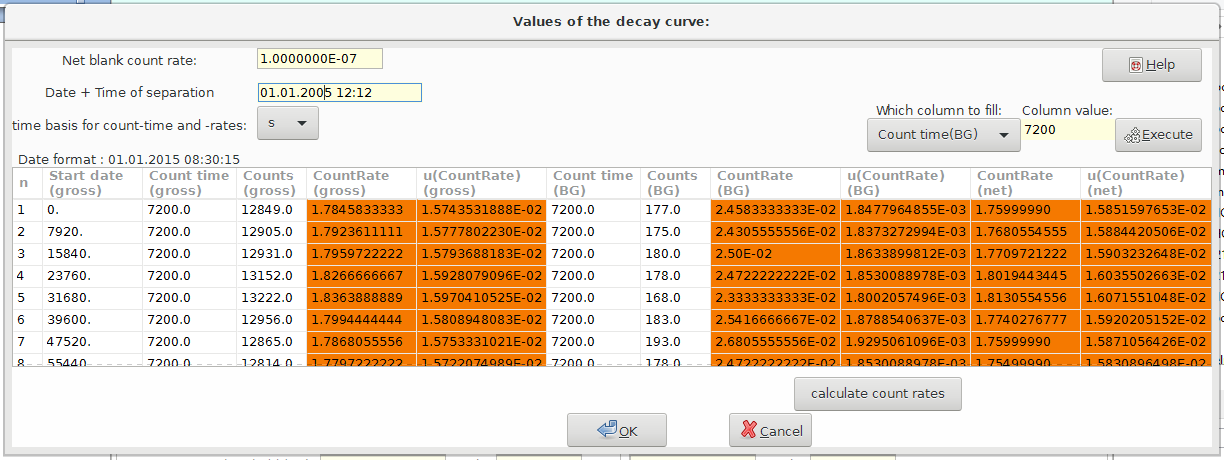
For the **input of the single measurement values of the decay curve** a table with 11 columns is available, the 11 columns of which have the following meaning; therein, only those columns shown in black color shall be filled in by the user, while the columns with red color are calculated by the program (see below).

|  |  |  |
| --- | --- | --- |
| Column-# | Name | Meaning |
| *Measurements of the gross counting rate:* | | |
| 1 | Start date | Date/time of the start of the *k*-th measurement (input format as shown above), or,  time duration between Y-90/Sr-90 separation and the start of the *k*-th measurement (as a number with decimal point); the program realizes automatically which of the two possibilities is used. |
| 2 | Count time | counting time of the *k*-th measurement |
| 3 | Counts | gross counts of the *k*-th measurement |
| 4 | CountRate | calculated gross counting rate |
| 5 | u(CountRate) | uncertainty of the gross counting rate |
| *Measurements of the background counting rate:* | | |
| 6 | Count time | counting time of the *k*-th background measurement |
| 7 | Counts | counts of the *k*-th background measurement |
| 8 | CountRate | calculated background counting rate |
| 9 | u(CountRate) | uncertainty of the background counting rate |
| *Resulting values of the net counting rate:* | | |
| 10 | Net CountRate | calculated net counting rate |
| 11 | u(NetCountRate) | uncertainty of the net counting rate |

With the **button „Calculate count rates“** the values of gross counting rate, background counting rate and the net counting rate as well as their associated uncertainties are calculated in the columns 4/5, 8/9 and 10/11 (in the columns marked by red color). In this calculation, the value of the net blank counting rate *Rbl* is considered (see above).

**Simplifying input to columns 2, 6 und 7:**

A new option within this dialog has been inserted, by which within one of these columns a fixed value can be transferred to all necessary cells within the selected column.



**Tips for facilitating the input of values into the first three columns:**

**Import from ASCII file:** If the values to be entered into the first three columns exist as an ASCII file, also column-wise separated, each of those columns (one after another) can be marked and copied into the Windows clipboard and pasted into the corresponding UncertRadio dialog column (right Mouse button: “paste”). The mouse pointer needs only to be set into the uppermost cell of this column in advance. It is required that the ASCII editor has the option to select column blocks.

**Import from Excel file:** If the values to be entered exist already in an Excel file they can be copied with a mouse operation “copy and paste” via Windows clipboard to the corresponding cell area in the UncertRadio dialog. In this case, all three columns can be copied simultaneously in one step.

Important for both variants: After having inserted a block into the uppermost cell, the **data are spread into the column cells with using the Enter key**.

Further information about [**Editing of tables**](#URH_TABTRICKS_EN)

## 7.11 Dialog Definition of the decay curve model

### 7.11.1 One single output quantity defined

The following picture gives an overview of the structure of the dialog.

|  |
| --- |
|  |

**One single output quantity defined**

In this dialog the [**described multi-linear least squares method**](#URH_LSQ_EN) is presented for the case of more complex Y-90 decay-curves (net counting rates)

The three terms describe different radionuclide decay contributions to the net counting rates and are dependent on the time *t* but also on the counting time(s).

For the case of a Y-90 decay curve the meaning of the three terms is as follows:

|  |  |
| --- | --- |
|  | represents the decay of the Y-90 component; (in ) is the Y-90 counting rate contribution at the time of Y/Sr separation which is to be fitted; |
|  | represents the (practically) constant counting rate contribution from an impurity due to long-lived unidentified radionuclide (half-live assumed to be infinitely, is then set equal to 1). An example of this is Th-234; if this known or identified its half-live Hwzlong can be specified (in the TAB “Values, uncertainties“); (in ) gives the size of this contribution (to be fitted). |
| : | represents the decay of the relatively short-lived Ac-228 which can occur as impurity in the Y oxalate source; this term may also be used if the presence of short-lived contaminations of the counting source by short-lived Radon decay products; (in ) gives the size of this contribution; |

The user can choose whether the second and/or the third term shall be used in the model.

Furthermore, it may be chosen whether the fitting shall be done with statistical weighting with the inverse variances of the net counting rates - or non-weighted. The internal use of covariances of measured net counting rates may be de-selected for testing.

The type of the fitting procedure can be selected from four variants: WLS, PLSQ, PMLE und WTLS. The more complex but more time-consuming total least squares procedure (WTLS) is able to take uncertainties of the values and covariances between them into account. See also: [Chi-square options](#_Chi-square_options)

A checkbox within this dialog allows to select instead of the simpler least squares analysis (WLS) the more complex but also more time-consuming general least squares procedure (WTLS). The latter is able to consider directly also uncertainties of the values and covariances between them.

In most cases, the term functions Xi(t) between different measurements only differ by the time difference to the time of the Sr/Y separation (parameter t: tstart). Therefore, these functions Xi(t) needed to be given only once, for the first measurement (in case of more than one counting channel: for the first measurement of each counting channel).

However, values of the counting efficiencies contained therein may now differ from measurement to measurement. In this case, which can be selected by a new checkbox, all term functions for each measurement have to be input explicitly. Look at the new example project Sr89-Sr90\_IAEA\_AQ-27\_2013\_V2\_EN.txp which demonstrates this.

In a text field of the dialog the equations for the three functions can be defined. For the above example of the analysis of a Y-90 decay-curve they have to be defined as follows:

X1 = (1. - exp(-log(2.)\*tmess/HwzY90)) / (log(2.)\*tmess/HwzY90) \* exp(-log(2.)\*tstart/HwzY90)

X2 = (1. - exp(-log(2.)\*tmess/Hwzlong)) / (log(2.)\*tmess/Hwzlong) \* exp(-log(2.)\*tstart/Hwzlong)

X3 = (1. - exp(-log(2.)\*tmess/HwzAc228)) / (log(2.)\*tmess/HwzAc228) \* exp(-log(2.)\*tstart/HwzAc228)

Here, tmess and tstart are counting time and the time durations between Y-90/Sr-90 separation and the starts of the measurements, respectively. In fact, both variables are arrays and their values for each single measurement may be defined in a special dialog. Hwzxxx are the radionuclide specific half-lives. **The formulae given above take radioactive decay during the measurements into account**.

With using the new decay function fd() the above equations are shorter:

X1 = fd(tstart, tmess, log(2)/HwzY90)

X2 = fd(tstart, tmess, log(2)/Hwzlong)

X3 = fd(tstart, tmess, log(2)/HwzAc228)

### 7.11.2 Extension to two or three output quantities

**Extension to two or three output quantities**

By the simultaneous measurement of for instance Sr-90, Sr-89, and may be also of Sr-85, with a LSC counter the contributions of these radionuclides to the counting rates in two or three counting channels (energy regions), designated as A, B and C, are determined. Further details may be taken from the report AKU (2008), which is dealing with modern methods of the Sr-89/Sr-90 determination.

The number of counting channels (nchs) can be selected in the dialog. In the present case up to three decay curves , and have to be considered. Then up to three model equations are used instead of only one:

Now, the fitting parameters represent activities instead of counting rates in the first example; see above: one single output quantity. These are automatically inserted by the program into the list of symbols as new symbols FITP1, FITP2 and FITP3.

Within the program, this case is reduced to that of one single decay curve by concatenating the three fields of counting rates (in the order A, B and C). The same applies to the independent decay functions.

Counting rates:

Decay functions: Input within the dialog field for the terms:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

For a better understanding of this application, the reader is referred to the example project *DWD-LSC-3kanal-V2.txp*, which corresponds to a quite detailed presentation of the equations in the report AKU (2008; page 160). For this application, the nine decay functions are defined as follows:

X1 = eSr89A \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X2 = eSr90A \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) +eY90A \* &

lamY90/(tmess\*(lamY90-lamSr90)) \*( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\* &

tmess)-1.)+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X3 = eSr85A \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

X4 = eSr89B \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X5 = eSr90B \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) +eY90B \* &

lamY90/(tmess\*(lamY90-lamSr90)) \*( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\* &

tmess)-1.)+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X6 = eSr85B \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

X7 = eSr89C \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X8 = eSr90C \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) +eY90C \* &

lamY90/(tmess\*(lamY90-lamSr90)) \*( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\* &

tmess)-1.)+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X9 = eSr85C \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

Herein, decay constants lamNuclide instead of half-lives are used. The symbols beginning with e represent for the considered radionuclides their detection probabilities in the counting channels A, B and C.

With using the new decay function fd() the above equations are shorter:

X1 = eSr89A \* fd(tAS+tstart,tmess,lamSr89)

X2 = eSr90A \* fd(tAS+tstart,tmess,lamSr90) + &

eY90A \* lamY90/(lamY90-lamSr90) \* ( fd(tAS+tstart,tmess,lamSr90) - fd(tAS+tstart,tmess,lamY90) )

X3 = eSr85A \* fd(tAS+tstart,tmess,lamSr85)

X4 = eSr89B \* fd(tAS+tstart,tmess,lamSr89)

X5 = eSr90B \* fd(tAS+tstart,tmess,lamSr90) + &

eY90B \* lamY90/(lamY90-lamSr90) \* ( fd(tAS+tstart,tmess,lamSr90) - fd(tAS+tstart,tmess,lamY90) )

X6 = eSr85B \* fd(tAS+tstart,tmess,lamSr85)

X7 = eSr89C \* fd(tAS+tstart,tmess,lamSr89)

X8 = eSr90C \* fd(tAS+tstart,tmess,lamSr90) + &

eY90C \* lamY90/(lamY90-lamSr90) \* ( fd(tAS+tstart,tmess,lamSr90) - fd(tAS+tstart,tmess,lamY90) )

X9 = eSr85C \* fd(tAS+tstart,tmess,lamSr85)

The contribution of the fourth radionuclide, Y-90, which is in-growing from the decay of Sr-90, is attributed for by additional terms with eY90X in the expressions for X2, X5 and X8.

If the same calibration activity *A*cal of a radionuclide was used for the efficiency calibration of the two to three energy windows, these efficiencies are correlated. Their covariances, pair-wise given by

are to be inserted for each pair of energy windows N and M, separately for the present radionuclides, in the covariance grid under the TAB "Values, uncertainties".

### 7.11.3 Organizing of the Xi Functions

1. number of Xi formulae =

(number of counting channels) x (number of applied output quantities)

(applied output quantities: fitting parameters, for which “fit“ or “fixed“ was selected)

or

1. number of Xi formulae =

(number of measurements) x (number of Xi formulae) x

x (number of counting channels)

(if a formulae (Xi) is defined explicitly for each of the measurements)

The **prescribed sequence** of the Xi formulae is indicated in the following two examples. It formally corresponds to the sequence which would be obtained by an SQL statement

“ORDER BY counting channel, number of measurements, number of output quantity”.

Example 1: Case a): 2 counting channels, 4 measurements, 3 output quantities; the Xi(t) differ between measurements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| counting channel | measurement No. | index of X i (t) |  | running No. |
| 1 | 1 | 1 |  | 1 |
| 1 | 1 | 2 |  | 2 |
| 1 | 1 | 3 |  | 3 |
| 1 | 2 | 1 |  | 4 |
| 1 | 2 | 2 |  | 5 |
| 1 | 2 | 3 |  | 6 |
| 1 | 3 | 1 |  | 7 |
| 1 | 3 | 2 |  | 8 |
| 1 | 3 | 3 |  | 9 |
| 1 | 4 | 1 |  | 10 |
| 1 | 4 | 2 |  | 11 |
| 1 | 4 | 3 |  | 12 |
| 2 | 1 | 1 |  | 13 |
| 2 | 1 | 2 |  | 14 |
| 2 | 1 | 3 |  | 15 |
| 2 | 2 | 1 |  | 16 |
| 2 | 2 | 2 |  | 17 |
| 2 | 2 | 3 |  | 18 |
| 2 | 3 | 1 |  | 19 |
| 2 | 3 | 2 |  | 20 |
| 2 | 3 | 3 |  | 21 |
| 2 | 4 | 1 |  | 22 |
| 2 | 4 | 2 |  | 23 |
| 2 | 4 | 3 |  | 24 |

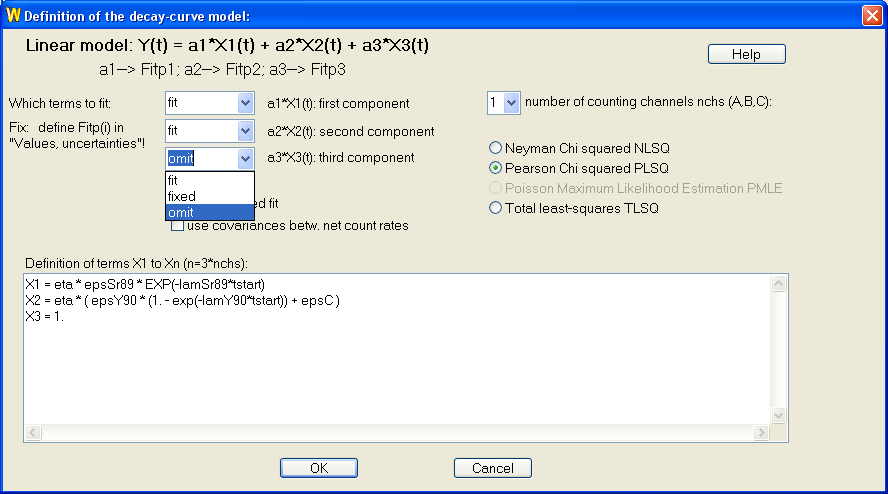
Example 2: case b), like example 1, but the Xi(t) do NOT differ between measurements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| counting channel | measurement No. | index of X i (t) |  | running No. |
| 1 | 1 | 1 |  | 1 |
| 1 | 1 | 2 |  | 2 |
| 1 | 1 | 3 |  | 3 |
| 2 | 1 | 1 |  | 4 |
| 2 | 1 | 2 |  | 5 |
| 2 | 1 | 3 |  | 6 |

### 7.11.4 One parameter excluded from fitting

**One parameter excluded from fitting**

There are three options for the terms :



The option “fixed“ became necessary by using a Sr-85 tracer within the Sr-90/Sr-89 analysis such that the Sr-85 activity is not subject to fitting but the chemical Sr yield is determined by an independent gamma measurement of Sr-85. This means that the Sr-85 contribution to the beta counting rates can be calculated separately. Select the Sr-85 option to “fix“ for enabling this case.

As the fitting routine cannot include the uncertainty of the Sr-85 activity (or its count rate), a special treatment was inferred. At first, for each of the gross count rate measurements the Sr-85 beta counting rate (and its uncertainty) is calculated and subtracted from the already available net counting rate .

The symbols are: the Sr-85 activity obtained by gamma-spectrometry; , the Sr-85 beta counting efficiency and the Sr-85 decay constant.

The function is determined by that equation, which the user defines as equation for X3 within the dialog for setting up a decay curve model. An example:

X3 = ASr85\_Gam \* eSr85 \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \*

exp(-lamSr85\*(tAS+tstart))

The associated fitting parameter (to be fixed) is internally set to the value 1. Collecting the input quantities , and into a vector , i.e. , allows to calculate the covariance matrix components needed for uncertainty propagation as follows:

diagonal values:

non-diagonal values:

The first term within the last equation does only occur if for calculating the net count rates always the same value of the background contribution is used (abbreviated here as ). The partial derivatives are calculated numerically.

The remaining unknown components for Sr-90 and Sr-89 are fitted to these Sr-85-corrected net counting rates (including their covariance matrix).

The symbols collected above into the vector must be included in the list of arguments of the call to Linfit, for example (note, that the equation for cSr85 is a dummy, i.e. only a place-holder):

cSr90 = Fitp1 \* PhiSr90

cSr89 = Fitp2 \* PhiSr89

cSr85 = Fitp3 \* 1

rd = Linfit(1, Rbl, **ASr85\_Gam**, **eSr85**, eSr90, eSr89, eY90, **lamSr85**, lamSr90,

lamSr89, lamY90, tmess, tstart )

phiSr90 = 1 / (etaSr\*Vol) \* exp(lamSr90 \* (tBS - tAS))

phiSr89 = 1 / (etaSr\*Vol) \* exp(lamSr89 \* (tBS - tAS))

X1 = eSr90 \* (1. - exp(-lamSr90\*tmess)) / (lamSr90\*tmess) \* exp(-lamSr90\*(tAS+tstart)) + &

eY90 \* lamY90/(tmess\*(lamY90-lamSr90)) \* &

( -exp(-lamSr90\*(tAS+tstart))/lamSr90\*(exp(-lamSr90\*tmess)-1.) &

+exp(-lamY90\*(tAS+tstart))/lamY90\*(exp(-lamY90\*tmess)-1.) )

X2 = eSr89 \* (1. - exp(-lamSr89\*tmess)) / (lamSr89\*tmess) \* exp(-lamSr89\*(tAS+tstart))

X3 = ASr85\_Gam \* eSr85 \* (1. - exp(-lamSr85\*tmess)) / (lamSr85\*tmess) \* exp(-lamSr85\*(tAS+tstart))

The equation for rd **shall be shortened to**:

rd = Linfit(1, Rbl, tmess, tstart )

## 7.12 Viewing the result from the LSQ fit to the decay curve

As a result, one obtains a table displayed by the programs internal editor (in this case no editing allowed) which has the following structure.

Result of decay curve analysis (with covariances): Method: PLSQ LinFit(t) = a1\*X1(t) + a2\*X2(t) + a3\*X3(t)

i t X1(t) X2(t) X3(t) NetRate rUnc. LinFit relDev uTest

(m) (cps) (%) (cps) (%)

-----------------------------------------------------------------------------------

1 433.00 0.83176 0.00000 0.00000 0.0017278 23.33 | 0.0016381 5.5 0.2

2 1633.00 0.67000 0.00000 0.00000 0.0013944 28.49 | 0.0013195 5.7 0.2 3 2833.00 0.53970 0.00000 0.00000 0.0009500 40.99 | 0.0010629 -10.6 -0.3

4 4033.00 0.43474 0.00000 0.00000 0.0009361 41.57 | 0.0008562 9.3 0.2

5 5237.00 0.34994 0.00000 0.00000 0.0006306 60.85 | 0.0006892 -8.5 -0.1

6 6437.00 0.28189 0.00000 0.00000 0.0006722 57.19 | 0.0005552 21.1 0.3

7 7637.00 0.22707 0.00000 0.00000 0.0006306 60.85 | 0.0004472 41.0 0.5

8 8837.00 0.18291 0.00000 0.00000 0.0002833 133.18 | 0.0003602 -21.3 -0.2

9 10037.00 0.15710 0.00000 0.00000 0.0004382 103.13 | 0.0003094 41.6 0.3

----------------------------------------------------------------------------------

LinFit: a1= 0.0019694 a2= 0.0000000 a3= 0.0000000 (given in cps !) ra1= 16.240 ra2= 0.000 ra3= 0.000 (given in % !) CHi2R= 8.481E-02 Prob= 0.000272 Prob= 0.000000 Prob= 0.000000 (t-test-signific. !)

The table columns are:

* No. of measurement i
* time duration between the Y-90/Sr-90 separation and the start time of the i-th measurement, is given in this example in m
* decay factors of the defined components dependent on t; in this example, only the decay of Y-90 is considered; 0.000 indicates that this decay component has not been defined/used
* , net counting rates, in 1/s
* relative standard uncertainty, in %
* , the value obtained from fitting the model for the net counting rate, in 1/s
* relDev., relative deviation , in %
* an u-test value, defined as . Absolute values being larger than 2, e.g. may indicate deviations from the model

After the table two rows follow showing the parameter values and their relative standard uncertainties obtained from the LSQ fitting. The first parameter, , gives the value of the Y-90 net counting rate being decay corrected to the time of the Y-90/Sr-90 separation. Chi2R is the value obtained for the reduced Chi-square.

**After closing** the editor window, **the model itself can again be edited** under the menu item “Edit->Decay curve->model of the decay curve” or with corresponding icon  if the first result is considered for improvement.

## 7.13 Calculation of the weighted mean and its standard uncertainty

The weighted mean of activities of the individual gamma lines is calculated according to the following equation:

(1)

The standard uncertainty of the weighted mean is calculated as follows:

(2)

This is termed as “internal standard” deviation. It only considers the uncertainties of the individual activities and is Bayes compliant.

It may, however, happen, that the individual values show deviations are larger than would be expected from the uncertainties . In order to consider also an additional uncertainty component due to these “external” influences, the so-called “external standard deviation” is often used being defined as follows:

(3)

In this context, the ratio is termed as Birge-ratio. For the statistical background of this ratio and of the terms of internal and external standard deviation it is referred the Appendix 2 in Kacker, Datla and Parr (2002).

Note, that this type of standard deviation of Eq. (3) is, however, no longer Bayes compliant.

The factors within the sums contained in the equations (1-3) represent statistical weights. These must be considered as being constant. Nevertheless, before they are applied, they are calculated from other variable values , which also contribute to the . If, after having calculated Eq. (1), followed by a numerical uncertainty propagation for with respect to the by using differences quotients follows, the values must not be modified. Under this constraint the uncertainty propagation for Eq. (1) directly yields the uncertainty given by Eq. (2). This condition is taken into account. The values may only be recalculated within the iterations for calculating the decision threshold and the detection limit, once per single iteration step.

**Internally, UncertRadio is calculating both, internal and external standard uncertainty but for further internal calculations it makes only use of the internal standard uncertainty.** The external standard deviation is shown only for information. The reason for doing that is that for the user it might be an useful information that the external standard deviation is considerably larger than the internal standard deviation. The latter can give advice about possible sources of errors which should be considered.

Covariances between peak efficiency values, taken from the same efficiency curve, are part of the covariance matrix **Ux** of the input values **x**:

**Important note**: According to Cox et al. (2006b), the equations for the weighted mean and its uncertainty given here can only be considered as “good” approximations, if covariances exist between the individual activity values. In such a case, instead, a least-squares procedure for the mean is to be applied. The corresponding procedure is described [below](#URH_GSPK1_LSQ_WMEAN_LSQ_EN).

## 7.14 Least-squares calculation of a weighted mean and its standard uncertainty

UncertRadio applies a matrix-based procedure described in section „[Mathematics of the linear LSQ curve fitting..](#URH_LLSQ_MATHE_EN)“, ***x*** = **A** ***y.***

The single activity values are taken as elements of the vector ***x***. The design matrix **A** = (1,1,…,1)T in this case has only one column the elements of which all are equal to1; ***y*** reduces to a vector consisting of only one value, the desired weighted mean.

For applying this procedure, the quadratic covariance matrix **Ux** is needed. UncertRadio assembles all the data into the corresponding algebraic elements and calculates also the required elements of the covariance matrix as already [described for the weighted mean](#URH_GSPK1_WMEAN_EN). The matrix **Uy** consists of only one element, the variance associated with weighted mean.

## 7.15 Approach of calculating Decision threshold and Detection limit for Gamspk1

The iterative calculation of the Decision threshold and the Detection limit is performed after the mean has been calculated from the single values . The iteration is done by variation of the mean where a varied value is . Then, all single values of the source activity are replaced by the new (fictive) value of . From the equation defining :

(1)

one obtains by inverting this equation

(2a)

and from this with the replacement an equation for the (fictive) net counting rates associated with the varied value :

(2b)

The aim is now to determine the uncertainties of the , then, via uncertainty propagation in accordance with Eq. (1), the uncertainties of the single activity values and, finally, with the chosen method for the mean to derive the uncertainty of the (iterated or fictive) mean value .

The following ansatz (a separation) is chosen for the uncertainties ([see also](#URH_GSPK1_WERTE_EN)):

(3)

Herein, only the first term is related directly to the contribution from the sample activity. The remaining terms represent uncertainty contributions of those parameters which characterize the background of the *i*-th gamma line including also a contribution from a “peak in the background”.

By using the last equation now with the equations (2b), (3) and (1) the (iterated or fictive) activities of the single gamma lines and their uncertainties can be determined as indicated already above. After these calculations those values are available which are necessary to go to the next iteration step and to test also for convergence of the iteration.

**Important note:**

„External“ influences may exist leading to calculated values of the single gamma lines which may exhibit a spreading which may be larger than to be expected from the uncertainties of single values.

This effect can be found with the **weighted mean** if the “external” is significantly larger than the “internal” standard uncertainty or the value of the “reduced Chi-square” significantly larger than one is.

In the case of the **arithmetic mean with additive correction** this may be inferred if the correction and particularly its uncertainty lead to a significant shift of the results compared to the arithmetic mean (uncorrected) or its uncertainty.

These influences usually are not considered in the evaluation model given by Eq. (1). Determining Decision threshold and Detection limit requires iteration of the activity values. The inversion of that equation, i.e. calculating the net count rates according to Eq. (1b) to be expected for a given (iterated) value of the activity, leads directly to the elimination of that “external” influence. Then, from the obtained net counting rates single activity values result from Eq. (1) having all the same identical value , i.e. their spreading is equal to zero!

This means for the calculation of Decision threshold and Detection limit that the external effect which may have been found from the primary evaluation of the output quantity in this latter case does not come into effect. Insofar, the usability of the external standard deviation with the weighted mean or with the NIST-2004 method is low, at least regarding Decision threshold and Detection limit.

## 7.16 Dialog Values from spectrum evaluation

Within a table in this dialog for each of the gamma lines measured values of the used quantities (symbols) and their uncertainties can be input line-by-line.

It is emphasized that it is expected in the case of a naturally occurring radionuclide that from the net counting rate also the net counting rate of the corresponding background peak has already been subtracted. **This must be considered also in advance when estimating the uncertainty of the net counting rate.** Under this assumption the input of the net counting rate of the background peak is not necessary.

It is assumed that all necessary values and their uncertainties can be taken from the evaluation report produced by the gamma spectrometry software.

The following picture gives an overview of the structure of the dialog.

|  |
| --- |
|  |

The measured values are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol names**  **in the dialog:** | **Meaning:** |  | **Symbols:** |
| Rnet | net counting rate of the gamma line  at energy , in |  |  |
| RT | counting rate of the integrated Compton-background in the region 1,7×Fwhm of peak *i* at energy , in |  |  |
| Rbg | net counting rate of a peak at energy in a separately measured background spectrum, in |  |  |
| effi | full energy peak efficiency at energy |  |  |
| pgamm | gamma emission probability of the line *i* |  |  |
| f\_att | self-attenuation correction for energy ;  it is used in its multiplicative form; |  |  |
| f\_coin | coincidence summing correction of  the line at energy ; it is used in its multiplicative form; |  |  |

The units of the net counting rates can be given in cps () or in cpm ().

The measured values have to be inserted as absolute values, not as relative values. For inserting their associated uncertainties, the radio buttons allow to choose between values in % or absolute values.

**Note:** The values for effi and pgamm can be supplied as absolute values only.

The activities *A*i of the individual lines are calculated as follows:

Furthermore, the uncertainties of the net count rates are calculated by the program according to the following equation:

Herein, is a factor which depends on how the net counting rate *R*ni of the peak has been evaluated. In the case of the “classical” total peak area (TPA) method, it is given by:

*b* is the width of the peak at its base, e.g. *b*=1,7xFwhm, and *L* is the number of channels which are used on both sides of the peak for determining the count rate of the background continuum.

If, however, *R*ni is determined by the method of peak fitting, may be approximated by a fixed value being „slightly larger than 1“. Then, this factor depends on the method used for peak fitting; it may be estimated by some sort of “calibration” calculations.

Within the upper part of the dialog a radio list field allows to **choose the type of mean** between:

* [weighted mean](#URH_GSPK1_WMEAN_EN),
* [mean by a weighted least-squares method](#URH_GSPK1_LSQ_WMEAN_EN).

Furthermore, a number field in this dialog allows inputting the value of the factor ; the use of efficiency covariances can be selected or de-selected.

The gamma energies and the net counting rates and their standard uncertainties must be entered directly in the table.

In the first column of the table one can select or de-select individual gamma lines.

## 7.17 View of the result from calculating a mean with Gamspk1

**For the weighted mean of the single line activities** one obtains the following interim report for the case of a measurement of Co-60 on a HPGe detector:

----------------------------------------------------------------------

(1 + b/2L) equivalent factor for Compton BG rate: 1.120

Individual peak data:

(pgamm\*fcoin is a measure for the importance of the line!)

i E PNRate epsPeak pgamm fatt fcoin (pgamm\*fcoin)

keV cps %

----------------------------------------------------------------------------------

1 1173.20 5.699E-03 0.7790 0.99850 1.0000 1.0615 1.0599 values

2.71 1.6789 0.03000 1.0000 1.3810 u\_rels in %

2 1332.50 5.360E-03 0.7030 0.99986 1.0000 1.0641 1.0640 values

2.76 1.6245 0.00060 1.0000 1.3890 u\_rels in %

Results from individual peak activities:

A(i) = PeakNetRate(i) \* (fatt(i) \* fcoin(i)) / (epsPeak(i) \* pgamm(i))

i E(keV) Activity (Bq) rel.StdDev (%)

--------------------------------------------------

1 1173.20 7.7771E-01 3.61

2 1332.50 8.1147E-01 3.63

Evaluation of the weighted mean:

weighted mean = 0.79379

int. std. dev. of the mean = 2.23372E-02 ( 2.81 %) (Bayes compliant)

ext. std. dev. of the mean = 1.85227E-02 ( 2.33 %) (not Bayes compliant)

Chi-square = test value T = 0.68763

reduced Chi-square = 0.68763

significance (Chi-square > T) = 4.06973 %

Note: only the internal standard deviation will be used hereafter!

----------------------------------------------------------------------

In the first table of this report the input data are shown in shortened form without reproducing their uncertainties. In this example the corrections for coincidence summing (fcoinsu) are quite significant because of the well-type counting geometry. The product (pgamm \* fcoinsu) given in the last column of that table is a measure for the weighting of the individual gamma lines.

The second table of this report shows the activity values (in Bq) and their relative standard uncertainties (in %) calculated for the individual lines. What follows are the data obtained from calculating the weighted mean.

**For the mean obtained by applying weighted least-squares** (LSQ Mean) calculated from several peaks one obtains for the lower result-part shown above for the weighted mean:

(Note: in this case the activity variances of the two gamma lines are practically identical; thus, there are nearly no deviations between this method and that of the weighted mean.)

Evaluation of the weighted mean by least-squares:

weighted mean = 0.79358

std. dev. of the mean = 2.23360E-02 ( 2.81 %)

reduced Chi-square = 0.86694

----------------------------------------------------------------------

## 7.18 Obtaining MC distributions and statistics derived of it in detail

During the Monte Carlo simulation successively three different distributions are produced from which some statistics are derived. The latter are the arithmetic mean, the standard deviation of the distribution and especially certain quantiles.

At the beginning the (large) number N of simulated measurement values of the output quantity is defined. This defines one “run”. Several runs (number *r*; maximum 50) may be calculated which then allows statistical evaluations to be made of the statistics mentioned above; from the r-fold repetition of the simulation one gets for instance with the standard deviation of the *r* values of a statistic an indication of its uncertainty.

**Determining a quantile**

The MC procedure use previously was primarily designed for nearly symmetric distributions of the output quantity. With applying more and more the GUM Supplement 1 with asymmetric distributions of some specific input quantities may result in a rather asymmetric distribution of the output quantity. Extreme values, to be expected for the distribution, had to be fixed in the program in advance, which turned out to be a disadvantage.

All MC values of the output quantity are now stored in an array. Now, for every of the three above mentioned distributions MC simulation yields arrays of up to 2 000 000 MC values. From these arrays mean and standard deviation are easily calculated. Quantiles are estimated by a distribution-free method, which requires sorting the arrays first. Only for the purpose of graphical representation as a histogram, the MC values have to be sorted into 20 000 Bins („multi-channel“); the limits of the histograms are determined after the MC simulation – and not before. These modifications made the MC procedure code easier to handle.

1. **Bayesian estimates**

For the estimation of mean, standard deviation and lower and upper confidence limits according to the Bayesian method only the distribution truncated at zero is used, values below zero are not used and not considered for the total number of values. The confidence limits are estimated as lower and upper quantile of the distribution. The total number of values considered in one run is only about N/2.

Instead of symmetrical confidence limits it is possible to calculate a special pair of coverage limits which has the shortest distance (Bayesian coverage interval of shortest length). Test case: Neutron-Dose-Cox-2006\_EN.txp. The activation of this option does not lead to a significant increase of simulation duration.

Input parameters being associated with an asymmetric or rectangular distribution may result in a significant **asymmetry of the output quantity distribution**. In the decision threshold case this may cause some deviation of the mean value from zero.

At first, a deviation from an expected distribution symmetrically to zero is tested. This is done by comparing the mean value *y*1,deviating from zero, with the MC uncertainty of the mean, *u*(*y*1)/√*N* (see below). In the case of

**an iteration is performed for the decision threshold using the secant method**, which shifts the mean value closer to zero**.** If necessary, up to 12 steps are performed, each of them requiring a full MC simulation; this leads to increasing computational effort. If the above ratio does not fall below 0.10, that distribution with the lowest ratio is selected from those of the 12 steps, from which the decision threshold is then calculated. *y*1 can be reduced by typically one to two orders of magnitude; it may happen, however, that *y*1 is reduced by hardly more that a factor of 10.

1. **Decision threshold**

For the estimation of the Decision threshold a modification is applied which consists of setting the “true value” of the net counting rate (thereby also of the activity) equal to zero. This results in a distribution which to about 50 % has negative values. In this case, no truncation at zero is applied. The upper (1-) quantile of this distribution represents the simulated value of the Decision threshold.

1. **Detection limit**

The estimation of the Detection limit is the most time-consuming part of the simulation, because in this case the distribution is shifted several times by changing the mean of the distribution, each time after the total number *N* is reached. The aim of the step-wise shifting hereby is that the (lower) ß Quantile of this distribution has to come as close to the value of the Decision threshold (determined before, see above) as possible. This iteration procedure is based on a bi-section method but with dividing the interval by linear interpolation.

Negative values are explicitly considered in these calculations.

The motivation not to discard negative parts of the distributions for the estimation of Decision threshold and Detection limit comes from the fact that the primary result for the output quantity shall be directly comparable with Decision threshold. If the negative parts would be discarded, a transformation being in analogy to the truncation would have to be applied to the primary output quantity before such a comparison

To make the distributions non-negative in the case of the Bayesian estimates is possible after a comparison between primary output quantity and the Decision limit has been done and has led to the conclusion that the assumption of a non-zero activity value in the sample is true.

**Estimating the MC uncertainties of the characteristic values**

Performing a MC-Simulation with only one single run did up to now not give any estimates of the uncertainties associated the characteristic values. Furthermore, when using a small number of runs, the estimates obtained from the spreading within the runs also are not very reliable.

As the resulting MC distributions in most cases are normal distributions, uncertainties of characteristic values can be roughly estimated as follows (see e.g. Barlow, 1999):

|  |  |  |
| --- | --- | --- |
| characteristic value: | width parameter: | formula for the (absolute) uncertainty: |
| Value of output quantity |  |  |
| uncertainty |  |  |
| lower confidence limit |  |  |
| upper confidence limit |  | dito |
| decision threshold |  |  |
| detecion limit |  | ,  with |

(N: number of MC-simulated single measurements; ϕ(.) and Φ(.): density function and distribution function of the standard normal distribution, respectively; see Options dialog for the probabilities α, β and γ).

The MC uncertainties calculated according to this table are given as relative values in percent within the MC part under the TAB „Results“, but only in the case of one single run.

**Graphical presentations**

The distributions which are produced according to the three methods discussed above are displayed as histograms in a separate window while the simulation is running. They show each distribution accumulated from the *r* runs which stabilize after only few (of *r*) repetitions. In the case of the Detection limit the accumulated distribution is displayed after the *r* runs are terminated. The x-axis (abscissa) corresponds to values of the evaluated quantity shown in the title of a plot; the y-axis (ordinate) shows the probability.

Example of the separate window with the MC graphs:

|  |
| --- |
| MCplotfile-d_EN.png |

**Vertical green lines** in the graphs characterize, from the left to the right, the following values:

**distribution of: values:**

output quantity lower confidence limit, best estimate (Bayesian), upper confidence limit

Decision threshold its value

Detection limit Decision threshold, Detection limit

After each single MC run that **Gaussian curve is potted in blue color** which corresponds to the result of the analytical procedure.

Note: The separate window with the MC graphs is maintained after completion of the MC simulation calculations. This allows for additional inspection of data shown under the different TABs and for invoking a result report and to go then back to “Results” TAB with this window. If data or options were changed during this step having the consequence that the original assumptions underlying the MC simulation are no longer valid, this MC window will be closed. This is also done when again calculations in the TAB “Values, uncertainties” or calculations initiated by the change from TAB “Values, uncertainties” to the TAB “Uncertainty budget” are invoked.

## 7.19 The PMLE procedure for non-linear unfolding

### 7.19.1 Notes on the application in UncertRadio

The procedure for Poisson MLE (PMLE) requires that the dependent quantities in unfolding, e.g. gross counts of a decay curve, are Poisson-distributed. As the latter are applied as net count rates for the other methods, these net count rate values are converted to gross counts within the program. The gross counts of a decay curve

;

are considered as non-correlated. must have the same value for all measured count rates, otherwise the shape of the gross counts decay curve would be disturbed.

If for instance an Y-90 decay curve shall be fitted, the corresponding model equation for the net count rate representation

converts to the model of the PMLE fit:

This requires inferring a second fitting contribution with the fitting parameter and . The parameter represents the sum of a background and a blank value, it will be fitted by the PMLE procedure and may end up with a value which can deviate from the value known from measurement.

**Features:**

This procedure needs to have one fitting parameter more (for the background) than the other procedures used for fitting net count rate decay curves. In UncertRadio it can therefore be applied only if not more than two physical components are to be determined. Both physical components should have the property that their associated curve-shapes should be different from being constant or quasi-constant within time. If one of these components represents the contribution of a radionuclide with a rather large half-live, such that the decay curve fails to show a decrease, the fitting procedure cannot differentiate between this contribution and that of . In such a case the PMLE procedure cannot be used.

Furthermore, the PMLE procedure can at present not be applied, if another fitting parameter is used with the status „fixed“, as is the case for a Sr measurement where a Sr-85 tracer was added to the sample.

The PMLE procedure is not selectable in a case with too few measurements compared to the number of parameters to be fitted.

**Applying this procedure:**

The data within the dialog “Input of decay curve“ are to be handled in the same way as for the other fitting procedures; nothing changes there.

Has a project already been established for the use with e.g. the WLS procedure, the program has already sufficient information about the components to be fitted and is able to decide whether the above-mentioned criteria for applying PMLE are fulfilled. If the criteria are not fulfilled, the selection of the PMLE procedure in the model dialog “model of decay curve“ is disabled.

Is the PMLE procedure selectable within the model dialog, the procedure, if selected, is processed in the same manner as for the other procedures. Only within the view of the fit result (), gross count rates are displayed instead of net count rates.

### 7.19.2 Basic information about non-linear PMLE

For decay curves of gross counts, which are Poisson-distributed, the Poisson MLE (PMLE) fitting method is the better one compared to weighted least squares (WLS), if the count numbers are rather small. This is supported by corresponding test results given at the end of section 7.4.3. Unfortunately, only non-linear unfolding procedures can be applied. For this purpose, the Levenberg-Marquardt-method was applied as it is implemented in the Matlab routine Lm, published by H. P. Gavin (2022). The Matlab code was converted to Fortran and extended for the application to PMLE:

* PMLE with its special definition of the Chi-square (see section 7.4.3) requires a modification of the Levenberg-Marquardt (LM) matrix algebra,
* For stabilizing the fitting („penalized fitting“) an additional Chi-square term is included, which shall prevent the fitting parameters from moving too far from their start values during the non-linear iteration.

Referring to the nomenclature used by H. P. Gavin in his paper, the following table gives a short information about the extensions of his mathematical treatment implemented here. In the first column, the equation numbers used by Gavon are cited. The main difference between Levenberg-Marquard formulae (left column) and those for the PMLE modifications (right column) consists in the applied covariance matrices (left column) and and (right column). The additional term serves for stabilizing the fit. designates the Jacobi-Matrix of the first partial derivatives of the fitting function with respect to the parameters .

|  |  |  |
| --- | --- | --- |
| **Gl.** | **LM-procedure by Gavin, with stabilization included** | **LM-PMLE-procedure by Gavin, modified for PMLE and stabilisation** |
| (2) |  |  |
| (6) |  |  |
| (11) |  |  |
| (13) |  |  |
| (16) |  |  |
|  |  |  |

**Test with MC simulations**

The performance of the procedure shall be demonstrated for a Y-90 decay curve with 9 measurements (example project vTI-Y90-16748\_BLW\_V2\_EN.txp). This project was modified by increasing the gross count number of the 9th point from a shorter counting duration (28200 s) to the duration of the other points (72000 s). A separate Fortran routine was used for the necessary MC simulations. The model for PMLE fitting of the gross counts is given by (, parameter of the Y-90-contribution; : parameter für the sum UG of background counts and the net blank counts (72000 s):

means the Y-90 decay corrections for 9 times . For start values of counts , 9 “true” values of were calculated according to the latter equation. After replacing them by Poisson distributed random counts, these were fitted resulting in values b. This step was repeated times. This allowed to extract statistical information, separately for and :

true given start value,

meanp mean of the MC-values,

sdp MC standard deviation of the values estimating the dispersion,

meansd mean of the standard deviations estimated by the fitting routine for each of the fits

The sequence of steps described so far is repeated 8 times by dividing the start values by two each time. All this was executed three times, for the fitting methods PMLE (non-linear) and PLSQ and WLS (linear). The statistical data obtained are presented in the following table.

Overall, the given true values are quite well reproduced by meanp; for smaller count numbers only for the WLS procedure some deviations are observed. A good agreement of the values for sdp (“real dispersion”) and meansd (standard deviation estimated by the fitting procedure) means statistical consistency. It is observed that again the WLS procedure is less consistent for lower count numbers.

The evaluation of this test also demonstrates that the PMLE method yields lower but statistically consistent estimates of the parameters describing counts



## 7.20 Treatment of numbers of counts and count rates

The feature for non-normal distributed numbers of counts and count rates to be described now refers to Monte Carlo simulations according to ISO 11929-2019, part 2. According to part 1 of ISO 11929:2019, the input quantities in any case are assumed as normal-distributed or are attributed to this distribution by the principle of maximum entropy.

According to the GUM Supplement 1 (JCGM 101:2008), clause 6.4.11, for counted events, which are Poisson distributed and represent an input quantity , e.g., counted photons, the following step is recommended to be taken for determining the distribution of the input quantity. If  **events** are counted, **a Gamma distribution is assigned to the posterior of the quantity** by applying the Bayes theorem and using a constant prior. This is to be used as the distribution associated with :

for (1)

Mean and variance are and , respectively. This refers to numbers of counts.

For Poisson-distributed numbers of counts, a Gamma distribution is assigned In ISO 11929-2019 to the associated count rate , where a prior is used instead of a constant prior. Mean and variance are in this case given as und , respectively.

In UncertRadio, the described step is treated as follows. As already given by Eq. (1), the Gamma distribution is assigned to the number of counts by selecting the distribution type „(N+x) rule“ for (this corresponds to a prior ). By calculating the corresponding count rate , which requires an equation like , the count rate is also Gamma distributed. This also means, that a count rate to be treated in UncertRadio as Gamma distributed, always requires defining it by an equation like .

When measuring an activity, two variants are to be considered,

* a measurement with pre-selected counting duration (the registered number is randomly distributed, following a **Poisson distribution**), and
* a measurement with pre-selected numbers of counts (the counting duration is randomly distributed, following an **Erlang distribution**).

The Erlang distribution is addressed in the textbook by Knoll (Knoll, G.F., Radiation Detection and Measurement, 2nd edition, (John Wiley, NewYork,1989), pp. 96-99);

See also:

International Safety Research, Safety Support Series, 2013. Radiation Counting Statistics. Volume 1. Canada.

Pengra, D., 2008:

<http://courses.washington.edu/phys433/muon_counting/counting_stats_tutorial_b.pdf>

Pishro-Nik, H., Introduction to Probability:

<https://www.probabilitycourse.com/chapter11/11_1_2_basic_concepts_of_the_poisson_process.php>

.

**Comparing Erlang and Poisson distributions**

The two distributions are defined as follows, with designating the count rate parameter:

Poisson distribution (2)

Erlang distribution (3)

The Erlang distribution is a Gamma distribution for integer-valued . The two formulae (2 and 3) lead to a simple relation:

(4)

Applying the Bayes theorem with a prior to both distributions results in the same posterior distribution for the count rate , a Gamma distribution:

**Measurement with pre-set time:**

(5)

**Measurement with pre-set counts:**

(6)

By equating the second parts of the two equations (5) and (6), the simple relation of Eq. (4) is obtained again:

or

(7)

If another prior is used for the Poisson distribution, , again a Gamma distribution is obtained, but a different one: .

**In the case of pre-set counts** (counting duration variable), **the Erlang distribution must be assigned to** by selecting the distribution type “Npreset“ **for** . By an also required equation like , the Gamma distribution is thereby internally assigned to the count rate .

Example project: **PresetCounts\_EN.txp**

## 7.21 Treatment of physical units

Within equations for calculating the value of an output quantity, the input quantities are fully described by a value and by a unit. Sometimes, it is not fully considered that instead of basic units (such as kg, m, s) derived units (such as g, cm, min) are used. For calculating the output quantity value correctly, with a combination of basic units as its unit, scale factors for derived units have to be inserted in the equations. There are two concepts supporting this within UncertRadio:

* Application of Trigger variables used to explicitly introduce unit scaling factors in equations; the reader is referred to section 7.21.5. The scope of application of such triggers normally is wider than its use for scale factors.
* One can try to derive the unit of a dependent quantity from the units of its input quantities by computations. This method can be used within UncertRadio, at present mainly designed as a testing option for a project. The more detailed description of such a method is the main purpose of this section.

Therefore, the aim is to correctly define for a project the number values and units of its input quantities. This also includes the use of derived units. It should then be possible for the program, to derive the unit of the output quantity, with including the conversion to basic units of the output quantity.

For successfully going this way, it is necessary to put more value on a systematized application of units and their notation in text editors. Therefore, it is **important to describe units of input quantities most completely**. To arrive, for instance, in the example for measuring an activity, at the unit “Bq” as a part of the output quantity’s unit, the following is especially important:

* Don’t leave the unit field of a detection efficiency empty, but use the unit string “1/Bq/s“;
* Apply a unit like “Bq\*s/kg” for a calibration factor;
* For a chemical yield, if determined by weighing, the unit strings “g/g“ or “g/kg“ may apply.

Missing units of input quantities may prevent from calculating the output quantity unit correctly.

**Important:** Unfortunately, this automated way of calculating units is not compatible with applying trigger variables for scaling factors.

### 7.21.1 Collection of basic units and derived units

It is necessary to differentiate between basic units and derived units. For evaluating an UncertRadio project, the aim is to replace derived units by their basic units. Such a conversion requires to apply an associated conversion factor to values und uncertainties of the quantities.

The UncertRadio installation includes two CSV files, **unitsTable\_EN.csv** and **units\_other\_EN.csv**. These are shown below.

**unitsTable\_EN.csv:**

The meaning of the columns is:

column A: basic units;

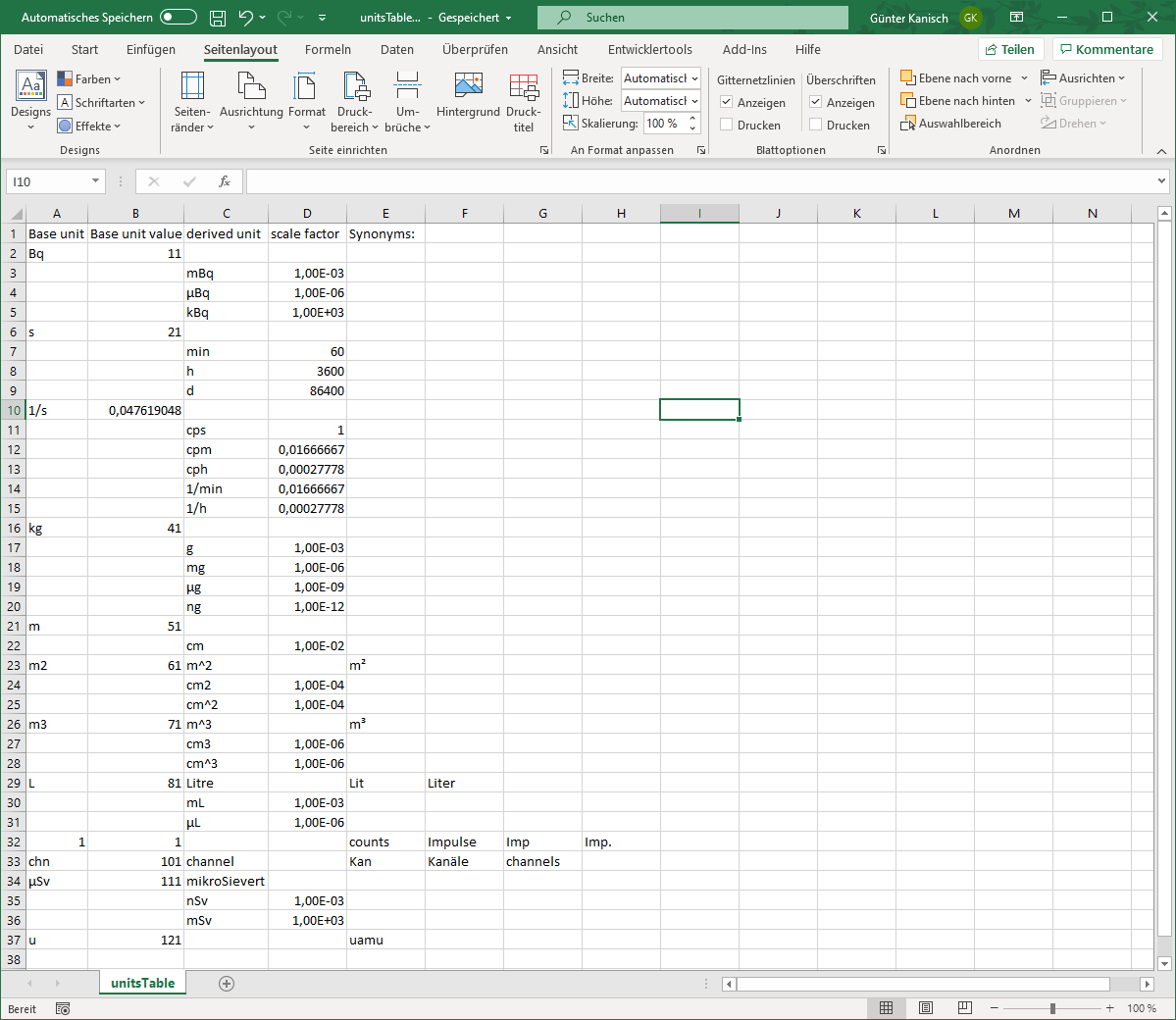
column B: a characteristic numerical value attributed to the basic unit;

column C: derived units;

column D: the scaling factor associated with the unit in column C;

column E and the following: within the row of the basic unit, several synonymous unit names can be inserted.

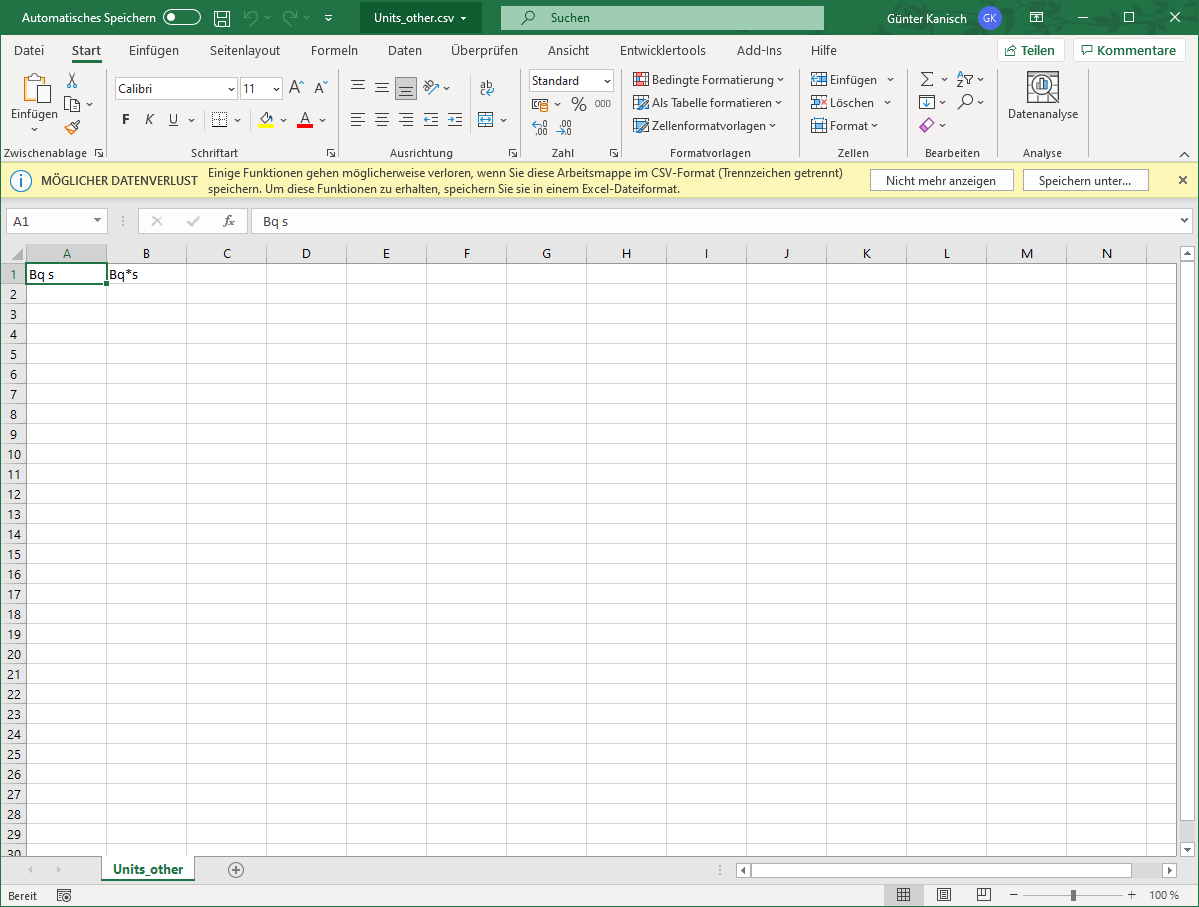
**Note:** As CSV files are language dependent, these two CSV files have been replaced by text files, which are shown next below the two CSV files.



*Note: The entries for m2 and m3 have been removed and transferred to Units\_other.txt*

**Units\_other\_EN.csv:**

This file contains only two columns, A and B. In column A, a unit name can be given in a notation preferred by a laboratory, column B gives the correct notation.



**The following two text files replace the CSV files**:

**unitsTable.txt**  (first part)**:**

base:Base unit;base#: base unit value; syn:synonym;derv:derived unit;conv: scale factor

base=Bq

base#=11.

derv=mBq

conv=1.00E-03

derv=µBq

conv=1.00E-06

derv=kBq

conv=1.00E+03

base=s

base#=21.

derv=min

conv=60.

derv=h

conv=3600.

derv=d

conv=86400.

base=1/s

base#=0.047619047619

derv=cps

conv=1.0

derv=cpm

conv=0.01666666666667

derv=cph

conv=0.000277777777778

derv=1/min

conv=0.01666666666667

derv=1/h

conv=0.000277777777778

base=kg

base#=41.

derv=g

conv=1.00E-03

derv=mg

conv=1.00E-06

**. . .**

**units\_other.txt**  (complete):

unit=Bq s

ubase=Bq\*s

unit=m2

ubase=m^2

unit=m³

ubase=m^3

unit=cm2

ubase=cm^2

unit=cm3

ubase=cm^3

The scaling factor associated with a counting duration is used (by inversion) for a count rate variable *R* and, in most cases, this factor is the same for *R* and for *u*(*R*), as long as the Poisson statistics is applicable. An exception is given by the gross count rate discussed in chapter 6.10, which is the sum of a binomial and a Poisson distributed quantity. For a calibration factor *w* or *phi*, which can be treated as a generalized product, the scaling factors for *w* or *phi* and for *u*(*w*) or *u*(*phi*) are the same.

### 7.21.2 Explaining the calculation of units of dependent quantities

For a dependent quantity the procedure is based on its equation given to UncertRadio. The right-hand side of this equation is an arithmetic expression (formula) of variable names. For calculating a unit, in a first step, the variable names are replaced by unit names as strings; a variable name “eps“ for a detection probability, e.g., is replaced by “(1/eps/s)“; the brackets shall assure that this expression, after insertion into the equation, is treated algebraically correct. In a later step, the unit parts contained in it, “Bq“ and “s“, are replaced by the characteristic numerical values “11“ and “21“ given in column B of the file unitsTable.csv.

Before basic units can used for calculations, the following modifications remain to be applied:

* the right-hand side of an equation with the number consists of some quantities numbered by ; the unit of the quantity with number can consist of one or more unit parts. The -th quantity has an index or address within the completed symbol table within UR.
* outside of arguments of function, all minus characters are replaced by plus characters; this shall assure that in case of a simple net count rate the difference of the unit values shall not become zero.
* For functions used inside a formula, like Log(), Exp() and Sqrt(), the variable names inside their arguments are replaced by unit names and later by their characteristic unit values. In this was, the argument of such a function gets a form which can be calculated numerically. For this purpose, a **second, simpler function parser is used UncertRadio, called seval**, which can calculate the formula string, if containing only numbers, directly, it does not operate on variables.
* If the argument of Log() (mostly 2) does not contain a variable with non-empty unit, i.e., a number, the expression Log(Argument) is set equal to 1.
* The unit of an input quantity with number can contain more than one unit parts, such that the unit represents a small formula. For a detection probability , the unit could be 1/mBq/min. The unit parts are converted to basic units and the associated conversion factors are combined in the same way to build the conversion of this input quantity: in the example, the scaling factor of the combined unit is: , if the desired unit shall be 1/Bq/s.
* To enable calculation, a unit string is build, for the example , from the characteristic unit values (see unitsTable.csv): “(1/11.0/21.0)“.
* Within an equation (for a dependent quantity) every single variable name contained in it (number ) is replaced by such a string. The scaling factor for the quantity associated with equation is determined from:

(1)

Herein, the string is the formula string of equation , in which the names of the symbols are replaced by the product converted to a string; is the formula string of equation , in which the names of the symbols are replace by converted to a string. Note: Messwert() denotes the array of measurement values (called MVals() in this test).

Example for (LAMSR \* TS \* 60^0) / (1. - EXP(-LAMSR \* TS \* 60^0)):

strgv1=(( 7.63000000E-10) \* ( 2.40000000E+04) \* 60^0) / (1. - EXP(-( 7.63000000E-10) \* ( 2.40000000E+04) \* 60^0))

strgv3=(( 7.63000000E-10) \* ( 4.00000000E+02) \* 60^0) / (1. - EXP(-( 7.63000000E-10) \* ( 4.00000000E+02) \* 60^0))

uconv(i) = 1.00000906

* It is assumed that the argument of an Exp function contains only quantities like a decay constant *lambda* (1/s) and a counting duration *t* (s). It is then allowed for this argument, that besides the characteristic unit values also scaling factors of , or and at the same time also associated factors like , build from Trigger variables, may occur. If then the overall argument value is not an integer value and is not equal to , or , a unit error is assumed; otherwise, the whole Exp(Argument) expression is set equal to 1.0 (Exp(Argument)=1).

Often, the Exp() expression occurs in the form of Form (1.0 – Exp()). The Minus sign in it is replaced by a Plus sign. If the analysis of the Exp() expression alone led to the result that it was set equal to 1, the whole (1.0 – Exp()) expression is set equal to 1. This still requires finding in the string the position of the left (opening) bracket.

* In the case of a sum in the argument of Sqrt(), e.g., for the variance of a net count rate with three terms with the unit (1/s^2), the value 3.0 \* (1/s^2) is expected for the argument. If, in this example, the factor 3.0 is not obtained, but a non-integer value, it can be assumed that at least one of the three terms carries a differing unit. If, however, an integer-valued factor is obtained, the whole sqrt expression can be replaced by the unit 1.

Following the replacement of an algebraic function expression for equation by substrings containing the characteristic unit values, a formula string should have been obtained (as a string RSeiteG2(i)), which can be evaluated by *seval*. The resulting numerical value is *Evalue*:

(2)

Arrived at this stage, it may have happened that some unities cancelled out.

The question then arises, how many – and which ones – basic units remain to contribute to *Evalue* (Eq. (2))? Therefore, in this formula string, the individual unit values, like “21.0“ for “s“ string, are replaced by their basic units ( as strings). Based on pairs “Basic unit name, unit value“, which are taken from the first two columns of the file unitsTable.csv, the more complex function parser *parsef* can be used for calculating partial derivatives of the formula with respect to the basic units. Only such individual units contribute to the unit of which show partial derivatives having (practically) non-zero values.

Now, when the set of participating unit parts is known, e.g., “Bq“, “s“ and “kg“, it has to be found out, which of them belong to the nominator or to the denominator of a generalized product. For these three unit parts, abbreviated now by a, b and c, the following possibilities have to tested:

(3)

The 8 possible combinations are tested numerically; if one of it results in the above-mentioned value *Evalue* (Eq. (2), the correct combination is found: e.g., “Bq\*s/kg“, if the value *Evalue* is equal to 11.0\*21.0/41.0.

### 7.21.3 Adjustments in the procedure

First, the units of all (independent) input quantities are replaced by basic units. If this implies a scaling factor not being one, the associate quantity value is (temporarily) multiplied with this factor.

In the next step, the units of the dependent quantities are calculated from the just treated input quantity units as described above.

The scaling factors of the dependent quantities are not derived from unit calculations. Instead, after the modification of the input quantities, they are calculated internally using the function Resulta without considering their units. Their associated unit scaling factors unit\_conv\_factor() are calculated thereafter, simply as ratios of the new quantity values and their previous values. Values of the uncertainties are then scaled by these factors unit\_conv\_factor().

### 7.21.4 Invoking the test of unit calculations

The calculation of units of dependent quantities cab be invoked under the menu item “Edit – test physical units“. The project should be developed such far that values are available under the TAB “Results”.

**Important**:

* The calculation of units implies the conversion to basic units. If other basic units are desired, the latter must be declared as basic units in the file unitsTable.csv. For example, if the unit kg shall be replaced by the unit g, make g to the basic unit and the kg to a derived unit.
* If the project contains a Trigger variable, invoking the unit test modus is prevented from invoking. The reason is, that a modified project saved directly by this test mode as a new file, normally does not work properly.
* If the test modus still indicates errors, this modus must be finished with explicitly using the close button of the Editor Tab. This leads to restoring the original status of program data.
* If no unit-related errors are shown, an addition button appears. It allows to save the modified state of the project as a new project file. This normally is necessary only if there are obvious deviations between the output quantity values.

The program executes the calculations according to chapter 7.21.2 and then displays in the program editor a comparison for the list of symbols.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

For dependent quantities (indexes ), the unit names given primarily by the user are replaced by “calculated“ unit names, which means that one must take care about the changed status of the project.

For the output of this test, UncertRadio calculates scaled values of measurement and associated standard uncertainties (*MVals\_scd* und *StdUnc\_scd*) as follows:

* The column „MVal\_scd/MVals\_org“ shows the obtained actual scaling factors unit\_conv\_factor().
* The values *Mvals\_scd* (dependent), derived with the function Resulta from the modified (independent) input quantity values.
* All values *StdUnc\_scd*, by scaling the previous values *StdUnc\_org* with the factors unit\_conv\_factor().

The output of the comparison test in the editor starts with a first error message, if the comparison between the value Evalue (Eq. (2)) and values from Eq. (3) (previous section) does not come to any agreement.

### 7.21.5 Introduction of Trigger variables

The program cannot give direct recommendations how to proceed if the test runs into an error. One reason can be a scaling factor which already exists in an equation like 60 for transforming “min“ into “s“. The impact by such already existing scaling factors can be reduced by using special switching variables (see chapter 2.2.6): for “Minutes (min)“ or for “Gram (g)“ already existing scaling factors 60 or 1/1000 are to be replaced as follows (alternatively, these factors are replaced by 1 and their unit is changed explicitly to “s” of “kg”):

60 🡪 60^min\_Trigger

1/1000 🡪 1/1000^kilo\_Trigger

These two special Trigger names are directly interpreted by UncertRadio; the value zero is assigned to them, if the Menu item „test physical units“ is applied, and prior to the test, they get the value 1.

The problem, that the expected unit of the output quantity, “Bq“ in the case of an activity, does not contain the substring “Bq“, but “1/s“ instead, can very often be solved by attributing the unit “1/Bq/s” to the detection probability.

**Test example: Janszen-Sr-89-Sr-90\_V2\_DE.txp**

The execution of the test stops with an error message. Then, the following changes are applied:

The first two equations:

a89 = As89 / (ms/1000.)

a90 = AsS90 / (ms/1000.)

are changed to:

a89 = As89 / (ms/1000.^kilo\_Trigger)

a90 = AS90 / (ms/1000.^kilo\_Trigger)

To obtain the desired unit “Bq/kg“, set the units of the four detection probabilities epsXXX to “1/Bq/s”.

The project file including these changes is available as **Janszen-Sr-89-Sr-90\_V3\_DE.txp**.

If a unit error is introduced, e.g., by changing in the symbol table under the TAB “Equations“ the unit of t2m0 from “s“ to “min“, then move the calculations forward to the TAB “Results“. An expected error message is then obtained when the unit test is called, in this case:

Error messages:

Eq. #=8 Error CLCU: Units in EXP argument do not match: seval=-60.0000000 arg(EXP)=-(1.0/21.0) \* ( 6.000000E+01\*(21.0))

Eq. #=4 RD89 = RSr - w\*RY\*f7 - Abl\*(etaSr \* epsSrSr89 \* f1): no unit found! Einvor=1 RSide=RSR-W\*RY\*F7-ABL\*(ETASR\*EPSSRSR89\*F1)

**Similar changes** had to be applied to the project **Moreno-Sr90\_IAEA-135\_EN.txp**.

The unit-strings “u (uamu)“ were replaced by “u“. Within the equations for the quantities a, w, f2 and epsSr, the two switching variables “kilo\_Trigger” or “min\_Trigger” were introduced. A unit “mg Sr“ was replaced by “mg“.

The project file including these changes is available as **Moreno-Sr90\_IAEA-135\_V2\_EN.txp**.

### 7.21.6 Experiences with the option for calculating units

Theo option of testing physical units could be integrated for the first time in the test with „QC-Batch-Test“, i.e., into the automatic evaluation run over all example projects. This allowed to identify some projects exhibiting specially selected units, for example,

Wuebbeler\_Ex2\*.txp (Ohm, Volt, Ampere)

Sterlinski\*.txp (n-activation, unit ng/g).

For the projects Kessel\*.txp and Calibration-of-weight-Cox\*.txp (calibration of masses), deviating factors of 1000 were observed. The project Calibration-of-weight-Cox\*.txp is still too complicated with respect to deriving units. For the projects Neutron-dose-Cox-2006\*.txp, changes of the resulting value by a factor of 100 per 36 are observed from changes in the input quantity units.

These projects are left as they are.

A real unit error was observed with the four projects sumEval\*V2\*.txp (DE+EN). For their version V2, at that time, the output quantity unit was set Bq/m²t, however, the associated area of 400 cm² had not been changed to 0,04 m². This error of the V2 version now soon became obvious by the new test modus. The new project version shown with the correct units, but output quantity values changed by the factor 100x100, has been saved, while still being in the test mode, as version V3 projects.

From the project Moreno-Sr90\_IAEA-135\_V2\*.txp, the triggers and the associated scaling factors of 60 or 1000 have been removed from the evaluation equations and the projects were then successful tested for unit calculations. With the processed derived units, the missing of the constant factors has been equalized. Value and uncertainty of the output quantity finally is left unchanged. The new project produced within the test modus is now released as a V3 version.

The same applied to the project Janszen-Sr-89-Sr-90\_V3\*.txp. The kilo\_trigger and the associated of the mass (g) were removed. The missing factor of 1000 in the equations has been equalized by the unit calculations, value and uncertainty of the output quantity are left unchanged. A new version V4 of these projects have been produced.